Welcome back...
Welcome back...
A metric space $X$, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$. Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.

(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.

(C) $X$ - vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) $X$ is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling. Hard problems. Easier to solve on trees. Dynamic programming on trees. Approximate metric on trees?
Metric spaces.

A metric space $X$, $d(i, j)$ where $d(i, j) \leq d(i, k) + d(k, j)$, $d(i, j) = d(j, i)$, and $d(i, j) \geq 0$.

Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.
(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.
(C) $X$- vertices in graph, $d(i, j)$ is shortest path distances in graph.
(D) $X$ is a set of vectors and $d(u, v)$ is $u \cdot v$. 
Metric spaces.

A metric space $X$, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.

(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.

(C) $X$- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) $X$ is a set of vectors and $d(u,v)$ is $u \cdot v$. 
A metric space $X$, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.
(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.
(C) $X$- vertices in graph, $d(i,j)$ is shortest path distances in graph.
(D) $X$ is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.
Metric spaces.

A metric space \( X, d(i,j) \) where \( d(i,j) \leq d(i,k) + d(k,j) \),
\( d(i,j) = d(j,i) \), and \( d(i,j) \geq 0 \).

Which are metric spaces?

(A) \( X \) from \( \mathbb{R}^d \) and \( d(\cdot, \cdot) \) is Euclidean distance.

(B) \( X \) from \( \mathbb{R}^d \) and \( d(\cdot, \cdot) \) is squared Euclidean distance.

(C) \( X \)- vertices in graph, \( d(i,j) \) is shortest path distances in graph.

(D) \( X \) is a set of vectors and \( d(u,v) \) is \( u \cdot v \).

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems.
Metric spaces.

A metric space \( X \), \( d(i,j) \) where \( d(i,j) \leq d(i,k) + d(k,j) \),
\( d(i,j) = d(j,i) \), and \( d(i,j) \geq 0 \).

Which are metric spaces?

(A) \( X \) from \( \mathbb{R}^d \) and \( d(\cdot, \cdot) \) is Euclidean distance.

(B) \( X \) from \( \mathbb{R}^d \) and \( d(\cdot, \cdot) \) is squared Euclidean distance.

(C) \( X \)- vertices in graph, \( d(i,j) \) is shortest path distances in graph.

(D) \( X \) is a set of vectors and \( d(u, v) \) is \( u \cdot v \).

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees.
A metric space $X$, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.

(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.

(C) $X$- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) $X$ is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees. Dynamic programming on trees.
Metric spaces.

A metric space $X$, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is Euclidean distance.

(B) $X$ from $\mathbb{R}^d$ and $d(\cdot, \cdot)$ is squared Euclidean distance.

(C) $X$- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) $X$ is a set of vectors and $d(u, v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees. Dynamic programming on trees.

Approximate metric on trees?
Approximate metric using a tree.

Tree metric:

- No distance shrinks. (dominating)
- Every distance stretches $\leq \alpha$ in expectation.

Distance 1 goes to $n - 1!$ Bummer.

Fix it up chappie!

For cycle, remove a random edge get a tree.

Stretch of edge: $n - 1 \times \frac{1}{n} + 1 \times \frac{n - 1}{n} \approx 2$ General metrics?
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
- Map $X$ into tree.
  - (i) No distance shrinks. (dominating)
  - (ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?
- Distance 1 goes to $n-1$!
- Bummer.
- Fix it up chappie!

For cycle, remove a random edge to get a tree.

Stretch of edge:
- $n - 1 
  - n \times 1
  - + \frac{1}{n} (n - 1)
  \approx 2

General metrics?
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights
- $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights
- $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Map $X$ into tree.
Approximate metric using a tree.

Tree metric:
- \( X \) is nodes of tree with edge weights
- \( d_T(i,j) \) shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Map \( X \) into tree.
- (i) No distance shrinks. (dominating)
Approximate metric using a tree.

Tree metric:
  $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Map $X$ into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
Approximate metric using a tree.

Tree metric:
$X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
Tree weights are geometrically decreasing.

Map $X$ into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches $\leq \alpha$

Map metric onto tree?
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Map $X$ into tree.
- (i) No distance shrinks. (dominating)
- (ii) Every distance stretches $\leq \alpha$

Map metric onto tree?
Approximate metric using a tree.

Tree metric:
\[ X \text{ is nodes of tree with edge weights } d_T(i,j) \text{ shortest path metric on tree.} \]

Hierarchically well separated tree metric:
Tree weights are geometrically decreasing.

Map \( X \) into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches \( \leq \alpha \)

Map metric onto tree?

Distance 1 goes to \( n - 1! \)
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Map $X$ into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches $\leq \alpha$

Map metric onto tree?

Distance 1 goes to $n - 1!$
Bummer.
Approximate metric using a tree.

Tree metric:

\[ X \text{ is nodes of tree with edge weights } d_T(i,j) \text{ shortest path metric on tree.} \]

Hierarchically well separated tree metric:

\[ \text{Tree weights are geometrically decreasing.} \]

Probabilistic Tree embedding.

Map \( X \) into tree.

(i) No distance shrinks. (dominating)

(ii) Every distance stretches \( \leq \alpha \) in expectation.

Map metric onto tree?

Fix it up chappie!

Distance 1 goes to \( n - 1! \)

Bummer.
Approximate metric using a tree.

Tree metric:
  \( X \) is nodes of tree with edge weights
  \( d_T(i,j) \) shortest path metric on tree.

Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
  Map \( X \) into tree.
    (i) No distance shrinks. (dominating)
    (ii) Every distance stretches \( \leq \alpha \)
        in expectation.

Map metric onto tree?
  Fix it up chappie!
  For cycle, remove a random edge

Distance 1 goes to \( n - 1! \)
Bummer.
Approximate metric using a tree.

Tree metric:

- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:

- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.

- Map $X$ into tree.
  - (i) No distance shrinks. (dominating)
  - (ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?

- Fix it up chappie!

For cycle, remove a random edge get a tree.
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
- Map $X$ into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?

Fix it up chappie!

For cycle, remove a random edge get a tree.
- Stretch of edge: $\frac{n-1}{n} \times 1$
Approximate metric using a tree.

Tree metric:

Let $X$ be the nodes of the tree with edge weights $d_T(i,j)$ being the shortest path metric on the tree.

Hierarchically well separated tree metric:

Tree weights are geometrically decreasing.

Probabilistic Tree embedding:

Map $X$ into the tree.

(i) No distance shrinks. (dominating)

(ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto a tree?

Fix it up chappie!

For cycle, remove a random edge to get a tree.

Stretch of edge:

$$\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1)$$
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights
- $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
- Map $X$ into tree.
  1. No distance shrinks. (dominating)
  2. Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?
- Fix it up chappie!

For cycle, remove a random edge get a tree.
- Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx$
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
- Map $X$ into tree.
  - (i) No distance shrinks. (dominating)
  - (ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?
- Fix it up chappie!

For cycle, remove a random edge get a tree.

- Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx 2$
Approximate metric using a tree.

Tree metric:
- $X$ is nodes of tree with edge weights $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
- Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
- Map $X$ into tree.
  - (i) No distance shrinks. (dominating)
  - (ii) Every distance stretches $\leq \alpha$ in expectation.

Map metric onto tree?

Fix it up chappie!

For cycle, remove a random edge get a tree.

Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx 2$

General metrics?
Probabilistic Tree embedding.

Map $X$ into tree.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).

Later: use spanning tree for graphical metrics.

The Idea: $\text{HST} \equiv \text{recursive decomposition of metric space}$.}

Decompose space by diameter $\approx \Delta$ balls. Recurse on each ball for $\Delta/2$. Use randomness in selection of ball centers. The $\approx$ diameter of the balls.
**Probabilistic Tree embedding.**

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Probabilistic Tree embedding.

Map $X$ into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:

HST $\equiv$ recursive decomposition of metric space.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Probabilistic Tree embedding.

Map $X$ into tree.

(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$. 
Probabilistic Tree embedding.

Map $X$ into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
Probabilistic Tree embedding.

Map $X$ into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in selection of ball centers.
Probabilistic Tree embedding.

Map $X$ into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
selection of ball centers.
The $\approx$ diameter of the balls.
Probabilistic Tree embedding.

Map $X$ into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expectation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of $X$ are leaves of tree.

Later: use spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
selection of ball centers.
the $\approx$ diameter of the balls.
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i,j) \geq 1\)
Algorithm

Algorithm: $(X, d)$, $\text{diam}(X) \leq D$, $|X| = n$, $d(i, j) \geq 1$

1. $\pi$ – random permutation of $X$. 
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   def subtree(S, \(\Delta\)):
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i,j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   ```python
   def subtree(S, \Delta):
       T = []
       if \Delta < 1 return [S]
       foreach i in \(\pi\):
           if \(i \in S\)
           \(B = \text{ball}(i, \beta \Delta)\);
           \(S = \frac{S}{B}\)
           \(T.append(B)\)
       return map (\(\lambda x: \text{subtree}(x, \frac{\Delta}{2})\), T);
   
   subtree(X, D)
   ```

Tree has internal node for each level of call.
Tree edges have weight \(\Delta\) to children.

Claim 1: \(d_T(x, y) \geq d(x, y)\).
When \(\Delta \leq d(x, y)\), \(x\) and \(y\) must be in different balls, so cut at lvl \(\Delta \geq d(x, y) / 2\).

\(\rightarrow d_T(x, y) \geq \Delta + \Delta \geq d(x, y)\)
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).

```python
def subtree(S, \Delta):
    T = []
    if \Delta < 1 return [S]
```
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   def subtree(S, \(\Delta\)):
      T = []
      if \(\Delta < 1\) return [S]
      foreach i in \(\pi\):
Algorithm

Algorithm: $(X, d)$, $\text{diam}(X) \leq D$, $|X| = n$, $d(i, j) \geq 1$

1. $\pi$ – random permutation of $X$.
2. Choose $\beta$ in $[\frac{3}{8}, \frac{1}{2}]$.
   def subtree(S, $\Delta$):
   T = []
   if $\Delta < 1$ return [S]
   foreach $i$ in $\pi$:
       if $i \in S$
Algorithm

Algorithm: $(X, d)$, diam$(X) \leq D$, $|X| = n$, $d(i, j) \geq 1$

1. $\pi$ – random permutation of $X$.
2. Choose $\beta$ in $[\frac{3}{8}, \frac{1}{2}]$.

   ```python
def subtree(S, $\Delta$):
    T = []
    if $\Delta < 1$ return [S]
    foreach i in $\pi$:
        if $i \in S$
            $B = \text{ball}(i, \beta \Delta)$
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i,j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   def subtree(S, \(\Delta\)):
   
   T = []
   if \(\Delta < 1\) return [S]
   foreach i in \(\pi\):
      if \(i \in S\)
         \(B = \text{ball}(i, \beta \Delta)\) ; \(S = S/B\)
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   ```python
def subtree(S, \Delta):
    T = []
    if \Delta < 1 return [S]
    foreach i in \(\pi\):
        if \(i \in S\):
            \(B = \text{ball}(i, \beta \Delta)\); \(S = S / B\)
            T.append(B)
    return map (\(\lambda x: \text{subtree}(x, \Delta / 2)\), T)
```

3. subtree\((X, D)\):

   Tree has internal node for each level of call.
   Tree edges have weight \(\Delta\) to children.

   Claim 1:
   \(d_T(x, y) \geq d(x, y)\).
   When \(\Delta \leq d(x, y)\), \(x\) and \(y\) must be in different balls, so cut at lvl \(\Delta \geq d(x, y) / 2\).

   \(\rightarrow d_T(x, y) \geq \Delta + \Delta \geq d(x, y)\).
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi - \) random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).

\[
\text{def subtree}(S, \Delta):
\]

\[
T = []
\]

if \(\Delta < 1\) return \([S]\)

foreach \(i\) in \(\pi\):

if \(i \in S\):

\[
B = \text{ball}(i, \beta \Delta) \ ; \ S = S/B
\]

T.append(B)

return map \((\lambda \ x: \text{subtree}(x, \Delta/2), T)\);
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi – \) random permutation of \(X\).
2. Choose \(\beta\) in \(\left[\frac{3}{8}, \frac{1}{2}\right]\).
   def subtree(S, \(\Delta\)):
       T = []
       if \(\Delta < 1\) return [S]
       foreach i in \(\pi\):
           if \(i \in S\)
               \(B = \text{ball}(i, \beta \Delta)\); \(S = S/B\)
               T.append(B)
       return map (\(\lambda x: \text{subtree}(x, \Delta/2), T\));
3. subtree(X, D)
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \(\left[ \frac{3}{8}, \frac{1}{2} \right]\).
   
   ```python
   def subtree(S, \Delta):
       T = []
       if \(\Delta < 1\) return [S]
       foreach i in \(\pi\):
           if \(i \in S\)
               \(B = \text{ball}(i, \beta\Delta)\); \(S = S/B\)
               T.append(B)
       return map (\(\lambda\ x: \text{subtree}(x,\Delta/2), T)\);
   ```
3. subtree\((X, D)\)

Tree has internal node for each level of call.
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i,j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   ```
   def subtree(S, \Delta):
       T = []
       if \Delta < 1 return [S]
       foreach i in \(\pi\):
           if \(i \in S\)
               \(B = \text{ball}(i, \beta \Delta)\); \(S = S/B\)
               T.append(B)
       return map (\lambda \ x: \text{subtree}(x, \Delta/2), T);
   ```
3. \text{subtree}(X, D)

Tree has internal node for each level of call. Tree edges have weight \(\Delta\) to children.
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).
   
   ```python
   def subtree(S, \Delta):
       T = []
       if \Delta < 1 return [S]
       for i in \pi:
           if \(i \in S\)
               \(B = \text{ball}(i, \beta \Delta) ; S = S/B\)
               T.append(B)
       return map (\lambda x: subtree(x, \Delta/2), T);
   ```
3. \(\text{subtree}(X, D)\)

Tree has internal node for each level of call. Tree edges have weight \(\Delta\) to children.

**Claim 1:** \(d_T(x, y) \geq d(x, y)\).
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)
1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([\frac{3}{8}, \frac{1}{2}]\).

```
def subtree(S, \Delta):
    T = []
    if \(\Delta < 1\) return [S]
    foreach \(i\) in \(\pi\):
        if \(i \in S\)
            \(B = \text{ball}(i, \beta \Delta); S = S/B\)
            \(T.\text{append}(B)\)
        return map (\(\lambda\ x: \text{subtree}(x, \Delta/2), T);\)
3. subtree(\(X, D\))
```

Tree has internal node for each level of call. Tree edges have weight \(\Delta\) to children.

**Claim 1:** \(d_T(x, y) \geq d(x, y)\).

When \(\Delta \leq d(x, y)\), \(x\) and \(y\) must be in different balls, so cut at lvl \(\Delta \geq d(x, y)/2\).
Algorithm

Algorithm: \((X, d), \text{diam}(X) \leq D, |X| = n, d(i, j) \geq 1\)

1. \(\pi\) – random permutation of \(X\).
2. Choose \(\beta\) in \([3/8, 1/2]\).
   
   ```python
def subtree(S, \Delta):
    T = []
    if \Delta < 1 return [S]
    foreach i in \(\pi\):
        if i \(\in\) S
            \(B = \text{ball}(i, \beta \Delta)\); \(S = S/B\)
            T.append(B)
    return map (\lambda x: subtree(x, \Delta/2), T);
```
3. subtree\((X, D)\)

Tree has internal node for each level of call. Tree edges have weight \(\Delta\) to children.

**Claim 1:** \(d_T(x, y) \geq d(x, y)\).

When \(\Delta \leq d(x, y)\), \(x\) and \(y\) must be in different balls, so cut at lvl \(\Delta \geq d(x, y)/2\).

\[
\rightarrow d_T(x, y) \geq \Delta + \Delta \geq d(x, y)
\]
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n)d(x, y)$. 
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n) d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. 
Analysis: idea

**Claim:** \( E[d_T(x, y)] = O(\log n) d(x, y) \).

Cut at level \( \Delta \) \( \rightarrow \) \( d_T(x, y) \leq 4\Delta \). (Level of subtree call.)
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n)d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level } \Delta]$?

Why should it be $d(x, y)\Delta$?

- smaller the edge the less likely to be on edge of ball.
- larger the $\Delta$, more room inside ball.
- random diameter jiggles edge of ball.

The problem?

Could be cut be many different balls.

For each probability is good, but could be hit by many.

random permutation to deal with this
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n)d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr$[cut at level $\Delta$]?

Would like it to be $\frac{d(x, y)}{\Delta}$.
Analysis: idea

**Claim:** $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level}\Delta]$?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$. 

Why should it be $d(x,y)/\delta$?

- smaller the edge the less likely to be on edge of ball.
- larger the $\Delta$, more room inside ball.

$\rightarrow$ $Pr[x,y \text{ cut by ball} | x \text{ in ball}] \approx d(x,y)\beta\Delta$

The problem?

Could be cut be many different balls.

For each probability is good, but could be hit by many.

random permutation to deal with this
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n) d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level } \Delta]$?

Would like it to be $\frac{d(x, y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta = D/2^i} (4\Delta) \frac{d(x, y)}{\Delta} = 4 \log D \cdot d(x, y)$.

Why should it be $\frac{d(x, y)}{\Delta}$?
Analysis: idea

Claim: \( E[d_T(x, y)] = O(\log n)d(x, y) \).

Cut at level \( \Delta \rightarrow d_T(x, y) \leq 4\Delta \). (Level of subtree call.)

\( Pr[\text{cut at level } \Delta] \)?

Would like it to be \( \frac{d(x, y)}{\Delta} \).

\( \rightarrow \) expected length is \( \sum_{\Delta=D/2^i}(4\Delta)\frac{d(x, y)}{\Delta} = 4\log D \cdot d(x, y) \).

Why should it be \( \frac{d(x, y)}{\Delta} \)?

smaller the edge the less likely to be on edge of ball.
**Claim:** \( E[d_T(x, y)] = O(\log n) d(x, y) \).

Cut at level \( \Delta \rightarrow d_T(x, y) \leq 4\Delta \). (Level of subtree call.)

\[ Pr[\text{cut at level } \Delta] \]

Would like it to be \( \frac{d(x,y)}{\Delta} \).

\[ \rightarrow \text{expected length is } \sum_{\Delta=2^i} \frac{d(x,y)}{\Delta} = 4 \log D \cdot d(x, y) \]

Why should it be \( \frac{d(x,y)}{\Delta} \)?

smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
Analysis: idea

**Claim:** \( E[d_T(x, y)] = O(\log n)d(x, y) \).

Cut at level \( \Delta \) \( \rightarrow \) \( d_T(x, y) \leq 4\Delta \). (Level of subtree call.)

\( Pr[\text{cut at level } \Delta] \)?

Would like it to be \( \frac{d(x, y)}{\Delta} \).

\( \rightarrow \) expected length is \( \sum_{\Delta=D/2}^{\infty} (4\Delta) \frac{d(x, y)}{\Delta} = 4\log D \cdot d(x, y) \).

Why should it be \( \frac{d(x, y)}{\Delta} \)?

smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.

random diameter jiggles edge of ball.
Analysis: idea

**Claim:** \( E[d_T(x, y)] = O(\log n)d(x, y) \).

Cut at level \( \Delta \) \( \rightarrow \) \( d_T(x, y) \leq 4\Delta \). (Level of subtree call.)

\[ Pr[\text{cut at level}\Delta]? \]

Would like it to be \( \frac{d(x,y)}{\Delta} \).

\[ \rightarrow \text{expected length is } \sum_{\Delta=\frac{D}{2^i}} (4\Delta) \frac{d(x,y)}{\Delta} = 4\log D \cdot d(x, y). \]

Why should it be \( \frac{d(x,y)}{\Delta} \)?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

\[ \rightarrow Pr[x, y \text{ cut by ball}| x \text{ in ball}] \approx \frac{d(x,y)}{\beta\Delta} \]
Analysis: idea

Claim: \( E[d_T(x, y)] = O(\log n) d(x, y) \).

Cut at level \( \Delta \) \( \rightarrow \) \( d_T(x, y) \leq 4\Delta \). (Level of subtree call.)

\( Pr[\text{cut at level}\Delta] \)?

Would like it to be \( \frac{d(x,y)}{\Delta} \).

\( \rightarrow \) expected length is \( \sum_{\Delta = D/2^i} (4\Delta) \frac{d(x,y)}{\Delta} = 4 \log D \cdot d(x, y) \).

Why should it be \( \frac{d(x,y)}{\Delta} \)?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

\( \rightarrow \) \( Pr[x, y \text{ cut by ball}|x \text{ in ball}] \approx \frac{d(x,y)}{\beta \Delta} \)

The problem?
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n)d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level}\Delta]$?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=\frac{D}{2}}^{i} (4\Delta) \frac{d(x,y)}{\Delta} = 4\log D \cdot d(x, y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

$\rightarrow Pr[x, y \text{ cut by ball}|x \text{ in ball}] \approx \frac{d(x,y)}{\beta\Delta}$

The problem?

Could be cut be many different balls.
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n) d(x, y)$.

Cut at level $\Delta \to d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level } \Delta]$?

Would like it to be $\frac{d(x, y)}{\Delta}$.

$\to$ expected length is $\sum_{\Delta} (4\Delta) \frac{d(x, y)}{\Delta} = 4 \log D \cdot d(x, y)$.

Why should it be $\frac{d(x, y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
   random diameter jiggles edge of ball.

$\to Pr[x, y \text{ cut by ball} | x \text{ in ball}] \approx \frac{d(x, y)}{\beta \Delta}$

The problem?

Could be cut be many different balls.
   For each probability is good, but could be hit by many.
Analysis: idea

**Claim:** $E[d_T(x, y)] = O(\log n)d(x, y)$.

Cut at level $\Delta \rightarrow d_T(x, y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level } \Delta]$?

Would like it to be $\frac{d(x, y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i} (4\Delta) \frac{d(x, y)}{\Delta} = 4\log D \cdot d(x, y)$.

Why should it be $\frac{d(x, y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

$\rightarrow Pr[x, y \text{ cut by ball} | x \text{ in ball}] \approx \frac{d(x, y)}{\beta \Delta}$

The problem?

Could be cut be many different balls.

For each probability is good, but could be hit by many.

random permutation to deal with this
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)
Analysis: $(x, y)$

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$

(Only consider cut by $x$, factor 2 loss.)

At level $\Delta$
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_{\Delta}\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}| x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}| x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.

Reumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),

Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if...

\(d(j, x) \leq \beta \Delta\)
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..
\[d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y)\]
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}| x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8, \frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..

\(d(j, x) \leq \beta \Delta\) and \(\beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\)
Analysis: $(x, y)$

Would like $Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$

(Only consider cut by $x$, factor 2 loss.)

At level $\Delta$

At some point $x$ is in some $\Delta$ level ball.
Renumber nodes in order of distance from $x$.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

$j$ can only cut $(x, y)$ if $d(j, x) \in [\Delta/4, \Delta/2]$ (else $(x, y)$ entirely in ball),
Call this set $X_\Delta$.

$j \in X_\Delta$ cuts $(x, y)$ if..
$d(j, x) \leq \beta \Delta$ and $\beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)$
$\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)]$. 
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball), Call this set \(X_{\Delta}\).

\(j \in X_{\Delta}\) cuts \((x, y)\) if..  
\[d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\]
\[\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].\]
occurs with prob. \(\frac{d(x, y)}{\Delta/8} = \frac{8d(x, y)}{\Delta}\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_{\Delta}\).

\(j \in X_{\Delta}\) cuts \((x, y)\) if.. \(d(j, x) \leq \beta \Delta\) and \(\beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\)

\(\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].\)

occurs with prob. \(\frac{d(x, y)}{\Delta/8} = \frac{8d(x, y)}{\Delta}\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..  
\[d(j, x) \leq \beta \Delta\text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\]  
\[\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].\]
occurs with prob. \(\frac{d(x, y)}{\Delta/8} = \frac{8d(x, y)}{\Delta}\).

And \(j\) must be before any \(i < j\) in \(\pi\)
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.

Reorder nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),

Call this set \(X_{\Delta}\).

\(j \in X_{\Delta}\) cuts \((x, y)\) if...

\[d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\]

\[\Rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].\]

occurs with prob. \(\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}\).

And \(j\) must be before any \(i < j\) in \(\pi\) \(\rightarrow\) prob is \(\frac{1}{j}\).
Analysis: $(x, y)$

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}$

(Only consider cut by $x$, factor 2 loss.)

At level $\Delta$

At some point $x$ is in some $\Delta$ level ball.

Renumber nodes in order of distance from $x$.

If $d(x, y) \geq \Delta / 8$, $\frac{8d(x, y)}{\Delta} \geq 1$, so claim holds trivially.

$j$ can only cut $(x, y)$ if $d(j, x) \in [\Delta / 4, \Delta / 2]$ (else $(x, y)$ entirely in ball),

Call this set $X_\Delta$.

$j \in X_\Delta$ cuts $(x, y)$ if..

\[
d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)
\]

$\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)]$.

occurs with prob. $\frac{d(x, y)}{\Delta/8} = \frac{8d(x, y)}{\Delta}$.

And $j$ must be before any $i < j$ in $\pi \rightarrow$ prob is $\frac{1}{j}$

$\rightarrow Pr[j \text{ cuts } (x, y)] \leq \left(\frac{1}{j}\right) \frac{8d(x, y)}{\Delta}$
Analysis: \((x, y)\)

Would like \(Pr[ x, y \text{ cut by ball}| x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \( x \), factor 2 loss.)

At level \(\Delta\)

At some point \( x \) is in some \(\Delta\) level ball.
Reumber nodes in order of distance from \( x \).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4,\Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..

\[d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)\]

\[\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].\]

occurs with prob. \(\frac{d(x, y)}{\Delta/8} = \frac{8d(x, y)}{\Delta}\).

And \(j\) must be before any \(i < j\) in \(\pi\) \(\rightarrow\) prob is \(\frac{1}{j}\)

\[\rightarrow Pr[j \text{ cuts } (x, y)] \leq \left(\frac{1}{j}\right)\frac{8d(x, y)}{\Delta}\]

\(d_T(x, y)\) if cut level \(\Delta\) is \(4\Delta\).
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}\)
(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Renumber nodes in order of distance from \(x\).

If \(d(x,y) \geq \Delta/8\), \(\frac{8d(x,y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..<br>
\[
d(j, x) \leq \beta \Delta \quad \text{and} \quad \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)
\]
\[
\rightarrow \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].
\]
occurs with prob. \(\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}\).

And \(j\) must be before any \(i < j\) in \(\pi\) \(\rightarrow\) prob is \(\frac{1}{j}\)

\(\rightarrow Pr[j \text{ cuts } (x, y)] \leq \left(\frac{1}{j}\right) \frac{8d(x,y)}{\Delta}\)

\(d_T(x, y)\) if cut level \(\Delta\) is \(4\Delta\).

\(\rightarrow E[d_T(x, y)] = \sum_{\Delta} \frac{D}{2^l} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x, y)\)
Analysis: \((x, y)\)

Would like \(Pr[x, y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x, y)}{\Delta}\)

(Only consider cut by \(x\), factor 2 loss.)

At level \(\Delta\)

At some point \(x\) is in some \(\Delta\) level ball.
Re-number nodes in order of distance from \(x\).

If \(d(x, y) \geq \Delta/8\), \(\frac{8d(x, y)}{\Delta} \geq 1\), so claim holds trivially.

\(j\) can only cut \((x, y)\) if \(d(j, x) \in [\Delta/4, \Delta/2]\) (else \((x, y)\) entirely in ball),
Call this set \(X_\Delta\).

\(j \in X_\Delta\) cuts \((x, y)\) if..
\[
d(j, x) \leq \beta \Delta \text{ and } \beta \Delta \leq d(j, y) \leq d(j, x) + d(x, y)
\]
\[
\implies \beta \Delta \in [d[j, x], d(j, x) + d(x, y)].
\]
occurs with prob. \(\frac{d(x, y)}{\Delta / 8} = \frac{8d(x, y)}{\Delta} \).

And \(j\) must be before any \(i < j\) in \(\pi\) \(\implies\) prob is \(\frac{1}{j}\)

\(\implies Pr[j \text{ cuts } (x, y)] \leq \left(\frac{1}{j}\right) \frac{8d(x, y)}{\Delta}\)

\(d_T(x, y)\) if cut level \(\Delta\) is \(4\Delta\).

\(\implies E[d_T(x, y)] = \sum_{\Delta = \frac{\Delta}{2^l}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x, y)\)
The pipes are distinct!

\[ E(d_T(x,y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x,y) \]
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_\Delta \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \).
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta=\Delta/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_\Delta \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_\Delta \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_\Delta \) is distinct from \( X_{\Delta/2} \).
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta=D/2^i} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_{\Delta} \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_{\Delta} \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_{\Delta} \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_{\Delta} \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq \sum_{j} \left( \frac{1}{j} \right) 32d(x, y) \]
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x, y) \]

Recall \( X_\Delta \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_\Delta \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta = \frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x, y) \leq \sum_{j} \left(\frac{1}{j}\right) 32d(x, y) \leq (32\ln n)(d(x, y)). \]
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta = D/2^i} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_{\Delta} \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_{\Delta} \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta = \frac{D}{2^i}} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq \sum_j \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq (32 \ln n) (d(x, y)) \]

**Claim:** \( E[d_T(x, y)] = O(\log n) d(x, y) \)
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta=D/2}^{\Delta} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_\Delta \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_\Delta \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta=D/2}^{\Delta} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq \sum_j \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq (32 \ln n) (d(x, y)) \]

**Claim:** \( E[d_T(x, y)] = O(\log n) d(x, y) \)

Expected stretch is \( O(\log n) \).
The pipes are distinct!

\[ E(d_T(x, y)) = \sum_{\Delta=D/2^i} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

Recall \( X_{\Delta} \) has nodes with \( d(x, j) \in [\Delta/4, \Delta/2] \)

“Listen Stash, the pipes are distinct!!”

Uh.. well \( X_{\Delta} \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x, y)) = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_{\Delta}} \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq \sum_j \left( \frac{1}{j} \right) 32d(x, y) \]

\[ \leq (32 \ln n) (d(x, y)). \]

**Claim:** \( E[d_T(x, y)] = O(\log n)d(x, y) \)

Expected stretch is \( O(\log n) \).

We gave an algorithm that produces a distribution of trees.
The pipes are distinct!

\[ E(d_T(x,y)) = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x,y) \]

Recall \( X_\Delta \) has nodes with \( d(x,j) \in [\Delta/4, \Delta/2] \)

"Listen Stash, the pipes are distinct!!"

Uh.. well \( X_\Delta \) is distinct from \( X_{\Delta/2} \).

\[ E(d_T(x,y)) = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left( \frac{1}{j} \right) 32d(x,y) \]

\[ \leq \sum_j \left( \frac{1}{j} \right) 32d(x,y) \]

\[ \leq (32 \ln n)(d(x,y)). \]

**Claim:** \( E[d_T(x,y)] = O(\log n)d(x,y) \)

Expected stretch is \( O(\log n) \).

We gave an algorithm that produces a distribution of trees.

The expected stretch of any pair is \( O(\log n) \).
Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$. 

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes
Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e=uv} c(e) d(l(u), l(v)) + \sum_{v} c(v, l(v))$$
Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e=(u,v)} c(e)d(\ell(u), \ell(v)) + \sum_{v} c(v, \ell(v))$$

Idea: find HST for metric $(X, d)$. 
Metric Labelling

Input: graph $G = (V, E)$ with edge weights $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$$\sum_{e = (u, v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.
Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.
Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e = (u, v)} c(e)d(l(u), l(v)) + \sum_{v} c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree,
Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e=(u,v)} c(e)d(l(u), l(v)) + \sum_v c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”,

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e=(u, v)} c(e) d(l(u), l(v)) + \sum_{v} c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”, constant factor.
Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels $(X, d)$, and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$$\sum_{e = (u, v)} c(e)d(l(u), l(v)) + \sum_{v} c(v, l(v))$$

Idea: find HST for metric $(X, d)$.

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”, constant factor.

$\rightarrow O(\log n)$ approximation.
And Now For Something...

Completely Different.
Example Problem: clustering.

- Points: documents, dna, preferences.
- Graphs: applications to VLSI, parallel processing, image segmentation.
Image example.
Image Segmentation

Normalized Cut: Find $S$, which minimizes $\frac{w(S, S^c)}{w(S)} \times \frac{w(S^c)}{w(S^c)}$.

Ratio Cut: minimize $\frac{w(S, S^c)}{w(S^c)}$ no more than half the weight. (Minimize cost per unit weight that is removed.)

Either is generally useful!
Image Segmentation

Which region?

Normalized Cut: Find $S$, which minimizes $w(S, \overline{S}) \times \frac{w(S)}{w(S) + w(\overline{S})}$.

Ratio Cut: minimize $w(S, \overline{S}) \times \frac{w(S)}{w(S) + w(\overline{S})}$. (Minimize cost per unit weight that is removed.)

Either is generally useful!
Image Segmentation

Which region?

Normalized Cut: Find $S$, which minimizes
$$w(S, S) \times w(S) \times w(S).$$

Ratio Cut: minimize
$$w(S, S) \times w(S),$$
$$w(S).$$

Either is generally useful!
Image Segmentation

Which region? Normalized Cut: Find $S$, which minimizes

$$\frac{w(S, \overline{S})}{w(S) \times w(\overline{S})}.$$
Image Segmentation

Which region? Normalized Cut: Find $S$, which minimizes

$$\frac{w(S, \bar{S})}{w(S) \times w(\bar{S})}.$$

Ratio Cut: minimize

$$\frac{w(S, \bar{S})}{w(S)},$$

$w(S)$ no more than half the weight. (Minimize cost per unit weight that is removed.)
Which region? Normalized Cut: Find $S$, which minimizes

$$\frac{w(S, \overline{S})}{w(S) \times w(\overline{S})}.$$ 

Ratio Cut: minimize

$$\frac{w(S, \overline{S})}{w(S)}.$$ 

$w(S)$ no more than half the weight. (Minimize cost per unit weight that is removed.)

Either is generally useful!
Edge Expansion/Conductance.

Graph $G = (V, E)$,
Graph $G = (V, E)$,
Assume regular graph of degree $d$. 

Edge Expansion/Conductance.

$h(S) = \frac{|E(S, V \setminus S)|}{\min(|S|, |V \setminus S|)}$,
$h(G) = \min_S h(S)$

$\phi(S) = \frac{n \cdot |E(S, V \setminus S)|}{d \cdot |S| |V \setminus S|}$,
$\phi(G) = \min_S \phi(S)$

Note $n \geq \max(|S|, |V \setminus S|) \geq n/2 
\Rightarrow h(G) \leq \phi(G) \leq 2h(S)$
Edge Expansion/Conductance.

Graph $G = (V, E)$,
Assume regular graph of degree $d$.
Edge Expansion.
Graph $G = (V, E)$,
Assume regular graph of degree $d$.

Edge Expansion.

$$h(S) = \frac{|E(S, V-S)|}{d \min |S|, |V-S|}, \ h(G) = \min_S h(S)$$
Edge Expansion/Conductance.

Graph $G = (V, E)$,
Assume regular graph of degree $d$.
Edge Expansion.
$$h(S) = \frac{|E(S, V-S)|}{d \min \{|S|, |V-S|\}}$$
$$h(G) = \min_S h(S)$$

Conductance.
$$\phi(S) = \frac{|E(S, V-S)|}{d |S||V-S|}$$
$$\phi(G) = \min_S \phi(S)$$

Note $n \geq \max(|S|, |V-S|) \geq n/2$ → $h(G) \leq \phi(G) \leq 2h(S)$
Graph $G = (V, E)$,
Assume regular graph of degree $d$.

**Edge Expansion.**

$$h(S) = \frac{|E(S, V - S)|}{d \min |S|, |V - S|}, \quad h(G) = \min_S h(S)$$

**Conductance.**

$$\phi(S) = \frac{n|E(S, V - S)|}{d |S||V - S|}, \quad \phi(G) = \min_S \phi(S)$$
Graph $G = (V, E)$,

Assume regular graph of degree $d$.

Edge Expansion.

$\text{Edge Expansion.}$

$h(S) = \frac{|E(S, V - S)|}{d \min(|S|, |V - S|)}$, $h(G) = \min_S h(S)$

Conductance.

$\text{Conductance.}$

$\phi(S) = \frac{n|E(S, V - S)|}{d|S||V - S|}$, $\phi(G) = \min_S \phi(S)$

Note $n \geq \max(|S|, |V| - |S|) \geq n/2$
Graph $G = (V, E)$,
Assume regular graph of degree $d$.

Edge Expansion.
$$h(S) = \frac{|E(S, V - S)|}{d \min |S|, |V - S|}, \quad h(G) = \min_S h(S)$$

Conductance.
$$\phi(S) = \frac{n |E(S, V - S)|}{d |S||V - S|}, \quad \phi(G) = \min_S \phi(S)$$

Note $n \geq \max(|S|, |V| - |S|) \geq n/2$
$$\rightarrow h(G) \leq \phi(G) \leq 2h(S)$$
Spectra of the graph.

\[ M = A/d \text{ adjacency matrix, } A \]
Spectra of the graph.

\[ M = A/d \text{ adjacency matrix, } A \]

Eigenvector: \( \nu - M\nu = \lambda \nu \)
Spectra of the graph.

\[ M = A/d \text{ adjacency matrix, } A \]

Eigenvector: \[ v - Mv = \lambda v \]

Real, symmetric.

Claim:

Any two eigenvectors with different eigenvalues are orthogonal.

Proof:

Eigenvectors: \( v, v' \) with eigenvalues \( \lambda, \lambda' \).

\[
 v^T M v' = v^T (\lambda' v') = \lambda' v^T v' = \lambda v^T v' = \lambda.
\]

Distinct eigenvalues \( \rightarrow \) orthonormal basis.

In basis: matrix is diagonal.

\[
 M = \begin{pmatrix}
 \lambda_1 & 0 & \ldots & 0 \\
 0 & \lambda_2 & \ldots & 0 \\
 \vdots & \vdots & \ddots & \vdots \\
 0 & 0 & \ldots & \lambda_n
\end{pmatrix}
\]
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

\[ \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix} \]
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).
Spectra of the graph.

\( M = A/d \) adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).

\[
\nu^T M \nu' = \nu^T (\lambda' \nu')
\]
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).

\[

\nu^T M \nu' = \nu^T (\lambda' \nu') = \lambda' \nu^T \nu'

\]
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).

\[ \nu^T M \nu' = \nu^T (\lambda' \nu') = \lambda' \nu^T \nu' \]

\[ \nu^T M \nu' = \lambda \nu^T \nu' \]
Spectra of the graph.

$M = A/d$ adjacency matrix, $A$

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: $v, v'$ with eigenvalues $\lambda, \lambda'$.

$v^T Mv' = v^T (\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v' = \lambda v^T v.$

Distinct eigenvalues

$\square$
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).

\[
\nu^T M \nu' = \nu^T (\lambda' \nu') = \lambda' \nu^T \nu'
\]

\[
\nu^T M \nu' = \lambda \nu^T \nu' = \lambda \nu^T \nu.
\]

Distinct eigenvalues \( \rightarrow \) orthonormal basis.
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( v - Mv = \lambda v \)

Real, symmetric.

**Claim:** Any two eigenvectors with different eigenvalues are orthogonal.

**Proof:** Eigenvectors: \( v, v' \) with eigenvalues \( \lambda, \lambda' \).

\[
v^T M v' = v^T (\lambda' v') = \lambda' v^T v' \]

\[
v^T M v' = \lambda v^T v' = \lambda v^T v. \]

Distinct eigenvalues \( \rightarrow \) orthonormal basis.

In basis: matrix is diagonal.
Spectra of the graph.

\[ M = A/d \] adjacency matrix, \( A \)

Eigenvector: \( \nu - M\nu = \lambda \nu \)

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: \( \nu, \nu' \) with eigenvalues \( \lambda, \lambda' \).

\[ \nu^T M \nu' = \nu^T (\lambda' \nu') = \lambda' \nu^T \nu' \]

\[ \nu^T M \nu' = \lambda \nu^T \nu' = \lambda \nu^T \nu. \]

Distinct eigenvalues \( \rightarrow \) orthonormal basis.

In basis: matrix is diagonal..

\[
M = \begin{bmatrix}
\lambda_1 & 0 & \ldots & 0 \\
0 & \lambda_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \lambda_n
\end{bmatrix}
\]
Action of $M$.

$v$ - assigns weights to vertices.
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d}\sum_{e=(i,j)}\nu_j$. 

Eigenvector with highest value?

$\nu = 1, \lambda_1 = 1$.

Claim:

For a connected graph $\lambda_2 < 1$.

Proof:

Second Eigenvector: $\nu_\bot 1$.

Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow\exists e = (i,j), \nu_i = x, x_j < x$.

Therefore $\lambda_2 < 1$.

Claim:

Connected if $\lambda_2 < 1$.

Proof:

Assign $+1$ to vertices in one component, $-\delta$ to rest.

$\nu_i = (M\nu_i) \Rightarrow$ eigenvector with $\lambda = 1$.

Choose $\delta$ to make $\sum_i \nu_i = 0$, i.e., $\nu_\bot 1$. 

Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value?
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d} \sum_{e=(i,j)} \nu_j$.

Eigenvector with highest value? $\nu = 1$. 

Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$. 
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d} \sum_{e=(i,j)} \nu_j$.

Eigenvector with highest value? $\nu = 1. \ \lambda_1 = 1$. 

$\rightarrow \nu_i$
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $\lambda_1 = 1$.

$\Rightarrow v_i = (M1)_i$
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1. \lambda_1 = 1.$

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1.$
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$. 
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp 1$. 
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d} \sum_{e=(i,j)} \nu_j$.

Eigenvector with highest value? $\nu = 1$. $\lambda_1 = 1$.

$\rightarrow \nu_i = (M1)_i = \frac{1}{d} \sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $\nu \perp 1$. Max value $x$.
Connected $\rightarrow$ path from $x$ valued node to lower value.
Action of $M$. 

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp 1$. Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), v_i = x$, $x_j < x$. 
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value?  $v = 1$.  $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp 1$.  Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j)$, $v_i = x$, $x_j < x$.  

\[ \begin{array}{c}
\vdots \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\downarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\}
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value?  $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp 1$. Max value $x$.
Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), \; v_i = x, \; x_j < x$.

\[ (Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) \]
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with \( \frac{1}{d} \sum_{e=(i,j)} v_j \).

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\Rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp 1$. Max value $x$.

Connected $\Rightarrow$ path from $x$ valued node to lower value.

$\Rightarrow \exists e = (i,j), v_i = x, x_j < x$.

\[
(Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x.
\]
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? \quad $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e \in (i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp 1$. Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), v_i = x, x_j < x$.

\[
(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x.
\]

Therefore $\lambda_2 < 1$. 
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d} \sum_{e=(i,j)} \nu_j$.

Eigenvector with highest value? $\nu = 1$. $\lambda_1 = 1$.

$\rightarrow \nu_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $\nu \perp 1$. Max value $x$.
Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), \nu_i = x, x_j < x$.

\[
\begin{array}{c}
\text{i} \quad \text{j} \\
\vdots \\
\text{x} \quad \leq \quad \text{x}
\end{array}
\]

$(M\nu)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. \qed
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d} \sum_{e\in(i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp \mathbf{1}$. Max value $x$. Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), v_i = x, x_j < x$.

\[
(Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x.
\]

Therefore $\lambda_2 < 1$. \hfill $\square$

**Claim:** Connected if $\lambda_2 < 1$. 

Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e \in (i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp 1$. Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists \ e = (i,j), \ v_i = x, \ x_j < x$.

\[
\begin{array}{c}
\text{(i)} \\
\text{(j)} \\
\text{---} \\
\text{x} \leq x
\end{array}
\]

$\rightarrow (Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$.  

**Claim:** Connected if $\lambda_2 < 1$.

**Proof:** Assign $+1$ to vertices in one component, $-\delta$ to rest.
Action of $M$.

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e \in (i,j)} 1 = 1$.

**Claim:** For a connected graph $\lambda_2 < 1$.

**Proof:** Second Eigenvector: $v \perp 1$. Max value $x$.
Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), v_i = x, x_j < x$.

\[
\begin{array}{c}
&i \\
\vdots & \downarrow \\
x & \leq x
\end{array}
\quad (Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x.
\]

Therefore $\lambda_2 < 1$. \qed

**Claim:** Connected if $\lambda_2 < 1$.

**Proof:** Assign $+1$ to vertices in one component, $-\delta$ to rest.

$x_i = (Mx_i)$
Action of $M$.

$\nu$ - assigns weights to vertices.

$M\nu$ replaces $\nu_i$ with $\frac{1}{d} \sum_{e=(i,j)} \nu_j$.

Eigenvector with highest value? \( \nu = 1 \). \( \lambda_1 = 1 \).

\[ \rightarrow \nu_i = (M1)_i = \frac{1}{d} \sum_{e \in (i,j)} 1 = 1. \]

**Claim:** For a connected graph \( \lambda_2 < 1 \).

**Proof:** Second Eigenvector: \( \nu \perp 1 \). Max value \( x \).

Connected \( \rightarrow \) path from \( x \) valued node to lower value.

\[ \rightarrow \exists e = (i,j), \nu_i = x, x_j < x. \]

\[ \begin{array}{c}
\vdots \\
 i \\
 \downarrow \\
 j \\
 \vdots \\
 x \\
 \leq x
\end{array} \quad \text{Therefore } \lambda_2 < 1. \]

\[ (M\nu)_i \leq \frac{1}{d} (x + x \cdots + \nu_j) < x. \]

**Claim:** Connected if \( \lambda_2 < 1 \).

**Proof:** Assign \(+1\) to vertices in one component, \(-\delta\) to rest.

\[ x_i = (Mx_i) \implies \text{eigenvector with } \lambda = 1. \]
Action of $M$. 

$v$ - assigns weights to vertices.

$Mv$ replaces $v_i$ with $\frac{1}{d} \sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = 1$. $\lambda_1 = 1$.

$\rightarrow v_i = (M1)_i = \frac{1}{d} \sum_{e \in (i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp 1$. Max value $x$.

Connected $\rightarrow$ path from $x$ valued node to lower value.

$\rightarrow \exists e = (i,j), v_i = x, x_j < x$.

$\begin{array}{c}
\vdots \\
\text{i} \\
\text{j} \\
\text{x} \\
\end{array}
\begin{array}{c}
\vdots \\
\text{x} \\
\end{array}

(Mv)_i \leq \frac{1}{d} (x + x \cdots + v_j) < x.$

Therefore $\lambda_2 < 1$. $

$\square$

Claim: Connected if $\lambda_2 < 1$.

Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.

$x_i = (Mx_i) \Longrightarrow$ eigenvector with $\lambda = 1$.

Choose $\delta$ to make $\sum_i x_i = 0$. 

Action of \( M \).

\( \nu \) - assigns weights to vertices.

\( \nu \) replaces \( \nu_i \) with \( \frac{1}{d} \sum_{e=(i,j)} \nu_j \).

Eigenvector with highest value? \( \nu = 1 \). \( \lambda_1 = 1 \).

\[ \nu_i = (M1)_i = \frac{1}{d} \sum_{e=(i,j)} 1 = 1. \]

Claim: For a connected graph \( \lambda_2 < 1 \).

Proof: Second Eigenvector: \( \nu \perp 1 \). Max value \( x \).

Connected \( \rightarrow \) path from \( x \) valued node to lower value.

\[ \rightarrow \exists e = (i,j), \nu_i = x, x_j < x. \]

\[ (M\nu)_i \leq \frac{1}{d} (x + x \cdots + \nu_j) < x. \]

Therefore \( \lambda_2 < 1 \).

Claim: Connected if \( \lambda_2 < 1 \).

Proof: Assign +1 to vertices in one component, \( -\delta \) to rest.

\[ x_i = (Mx_i) \implies \text{eigenvector with } \lambda = 1. \]

Choose \( \delta \) to make \( \sum_i x_i = 0 \), i.e., \( x \perp 1 \).
Rayleigh Quotient

\[ \lambda_1 = \max_x x^T M x \]

In basis, \( M \) is diagonal. Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \]

Tight when \( x \) is first eigenvector.

\[ \lambda_2 = \max_{x \perp 1} x^T M x \]

\( x \perp 1 \iff \sum_i x_i = 0 \).

Example: 0/1 Indicator vector for balanced cut, \( S \) is one such vector.

Rayleigh quotient is \( |E(S, S)| / |S| \).

Rayleigh quotient is less than \( h(S) \) for any balanced cut \( S \).

Find balanced cut from vector that achieves Rayleigh quotient?
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ xMx \]
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ xMx = \sum_i \lambda_i x_i^2 \]
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.
Rayleigh Quotient

$$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$$

In basis, $M$ is diagonal.

Represent $x$ in basis, i.e., $x_i = x \cdot v_i$.

$$x M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$$

Tight when $x$ is first eigenvector.

Rayleigh quotient.
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

\( x \perp 1 \)
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

\( x \perp 1 \iff \sum_i x_i = 0. \)
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]

\[ x \perp 1 \iff \sum_i x_i = 0. \]

Example: 0/1 Indicator vector for balanced cut, \( S \) is one such vector.
Rayleigh Quotient

\[
\lambda_1 = \max_x \frac{x^T M x}{x^T x}
\]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x\]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[
\lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}
\]

\( x \perp 1 \iff \sum_i x_i = 0 \).

Example: 0/1 Indicator vector for balanced cut, \( S \) is one such vector.

Rayleigh quotient is \( \frac{|E(S, \overline{S})|}{|S|} \).
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]

\( x \perp 1 \iff \sum_i x_i = 0 \).

Example: 0/1 Indicator vector for balanced cut, \( S \) is one such vector.

Rayleigh quotient is \( \frac{|E(S,S)|}{|S|} = h(S) \).
Rayleigh Quotient

$$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$$

In basis, $M$ is diagonal.

Represent $x$ in basis, i.e., $x_i = x \cdot v_i$.

$$x M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$$

Tight when $x$ is first eigenvector.

Rayleigh quotient.

$$\lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}.$$ 

$x \perp 1 \iff \sum_i x_i = 0$.

Example: 0/1 Indicator vector for balanced cut, $S$ is one such vector.

Rayleigh quotient is $\frac{|E(S,S)|}{|S|} = h(S)$.

Rayleigh quotient is less than $h(S)$ for any balanced cut $S$. 
Rayleigh Quotient

\[ \lambda_1 = \max_x \frac{x^T M x}{x^T x} \]

In basis, \( M \) is diagonal.

Represent \( x \) in basis, i.e., \( x_i = x \cdot v_i \).

\[ x^T M x = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x \]

Tight when \( x \) is first eigenvector.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]

\( x \perp 1 \leftrightarrow \sum_i x_i = 0 \).

Example: 0/1 Indicator vector for balanced cut, \( S \) is one such vector.

Rayleigh quotient is \( \frac{|E(S,S)|}{|S|} = h(S) \).

Rayleigh quotient is less than \( h(S) \) for any balanced cut \( S \).

Find balanced cut from vector that achieves Rayleigh quotient?
Cheeger’s inequality.

Rayleigh quotient.
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \cdot \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|} \)
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)
\[ \frac{\mu}{2} \]
Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \]
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} .\]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \]
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} \]
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu} \]
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: 
\[ h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \]

\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu} \]

Hmmm..
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|} \)
\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1. \)
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S,V-S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1 \).
\( h(G) \) large
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \bot 1} \frac{x^TMx}{x^Tx} \cdot \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, \lvert S \rvert \leq V/2} \frac{|E(S, V - S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1 \).

\( h(G) \) large \( \rightarrow \) well connected
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)
\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1. \)
\( h(G) \) large \( \rightarrow \) well connected \( \rightarrow \lambda_1 - \lambda_2 \) big.
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x} \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1 \).
\( h(G) \) large \( \rightarrow \) well connected \( \rightarrow \) \( \lambda_1 - \lambda_2 \) big.
Disconnected
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|} \)
\[ \frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1. \)
\( h(G) \) large \( \rightarrow \) well connected \( \rightarrow \) \( \lambda_1 - \lambda_2 \) big.
Disconnected \( \lambda_2 = \lambda_1. \)
Cheeger’s inequality.

Rayleigh quotient.

\[ \lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2 \).

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1 \).

\( h(G) \) large \( \rightarrow \) well connected \( \rightarrow \) \( \lambda_1 - \lambda_2 \) big.

Disconnected \( \lambda_2 = \lambda_1 \).

\( h(G) \) small
Cheeger’s inequality.

Rayleigh quotient.
\[ \lambda_2 = \max_{x \perp 1} \frac{x^T M x}{x^T x}. \]

Eigenvalue gap: \( \mu = \lambda_1 - \lambda_2. \)

Recall: \( h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|} \)

\[ \frac{\mu}{2} = \frac{1 - \lambda_2}{2} \leq h(G) \leq \sqrt{2(1 - \lambda_2)} = \sqrt{2\mu} \]

Hmmm..

Connected \( \lambda_2 < \lambda_1. \)
\( h(G) \) large \( \rightarrow \) well connected \( \rightarrow \) \( \lambda_1 - \lambda_2 \) big.
Disconnected \( \lambda_2 = \lambda_1. \)
\( h(G) \) small \( \rightarrow \) \( \lambda_1 - \lambda_2 \) small.
Easy side of Cheeger.

Small cut $\to$ small eigenvalue gap.
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut $S$. 
Easy side of Cheeger.

Small cut → small eigenvalue gap.

\[ \frac{\mu}{2} \leq h(G) \]

Cut \( S \). \( i \in S \): \( v_i = |V| - |S| \), \( i \in \overline{S}v_i = -|S| \).
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

\[ \frac{\mu}{2} \leq h(G) \]

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

\[ \sum_i v_i = |S||V| - |S|) - |S|(|V| - |S|) = 0 \]
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$$

$\rightarrow v \perp 1$. 
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S||V| - |S| - |S||V| - |S| = 0$$

$\rightarrow v \perp 1$.

$v^T v$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.
\[ \frac{\mu}{2} \leq h(G) \]

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S}v_i = -|S|$.
\[ \sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0 \]
$\rightarrow v \perp 1$.
\[ v^T v = |S|(|V| - |S|)^2 \]
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S}v_i = -|S|$.

$$\sum_i v_i = |S||V| - |S| - |S||V| - |S| = 0$$

$\rightarrow v \perp 1$.

$$v^Tv = |S||V| - |S|)^2 + |S|^2(|V| - |S|)$$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$. 

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp 1.$

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S| (|V| - |S|) - |S| (|V| - |S|) = 0$$

$\rightarrow v \perp 1$.

$$v^T v = |S| (|V| - |S|)^2 + |S|^2 (|V| - |S|) = |S| (|V| - |S|)(|V|).$$

$v^T M v$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S}v_i = -|S|$.  

$$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$$

$\rightarrow v \perp 1$.

$$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$$

$$v^T Mv = \frac{1}{d} \sum_{e=(i,j)} x_i x_j.$$
Easy side of Cheeger.

Small cut → small eigenvalue gap.

\( \frac{\mu}{2} \leq h(G) \)

Cut \( S \). \( i \in S : v_i = |V| - |S|, i \in \overline{S}v_i = -|S| \).

\[ \sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0 \]

→ \( v \perp 1 \).

\[ v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|). \]

\[ v^T Mv = \frac{1}{d} \sum_{e=(i,j)} x_i x_j. \]

Same side endpoints: like \( v^T v \).
Easy side of Cheeger.

Small cut → small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$$

$\rightarrow v \bot 1$.

$$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$$

$$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j.$$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$$

$\rightarrow v \perp 1$.

$$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$$

$$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j.$$

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \notin S v_i = -|S|.$

$$\sum_i v_i = |S||V| - |S||V| - |S| = 0$$

$\rightarrow v \perp 1.$

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j.$

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$$

$\rightarrow v \perp 1$.

$$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|).$$

$$v^T Mv = \frac{1}{d} \sum_{e=(i,j)} x_i x_j.$$

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

$$v^T Mv = v^T v - (2|E(S,S)||S|(|V| - |S|))$$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.  

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$  

$\rightarrow v \perp 1$.  

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.  

$v^TMv = \frac{1}{d}\sum_{e=(i,j)} x_i x_j$.  

Same side endpoints: like $v^Tv$.  

Different side endpoints: $-|S|(|V| - |S|)$  

$v^TMv = v^Tv - (2E(S,S)||S|(|V| - |S|)$  

$\frac{v^TMv}{v^Tv} = 1 - \frac{2|E(S,\overline{S})|}{|S|}$
Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$$\frac{\mu}{2} \leq h(G)$$

Cut $S$. $i \in S : v_i = |V| - |S|$, $i \in \overline{S}v_i = -|S|$.

$$\sum_i v_i = |S|||V| - |S|| - |S|||V| - |S|| = 0$$

$\rightarrow v \perp 1$.

$$v^Tv = |S|||V| - |S||^2 + |S|^2(||V| - |S||) = |S|||V| - |S||(|V|).$$

$$v^TMv = \frac{1}{d}\sum_{e=(i,j)} x_ix_j.$$

Same side endpoints: like $v^Tv$.

Different side endpoints: $-|S|||V| - |S||$

$$v^TMv = v^Tv - (2|E(S, S)||S|||V| - |S||)$$

$$\frac{v^TMv}{v^Tv} = 1 - \frac{2|E(S, \overline{S})|}{|S|}$$

$$\lambda_2 \geq 1 - 2h(S)$$
Easy side of Cheeger.

Small cut → small eigenvalue gap.

\[ \frac{\mu}{2} \leq h(G) \]

Cut \( S \). \( i \in S : \ v_i = |V| - |S|, \ i \in \overline{S} v_i = -|S|. \)

\[ \sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0 \]

\( \rightarrow v \perp 1. \)

\[ v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|). \]

\[ v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j. \]

Same side endpoints: like \( v^T v. \)

Different side endpoints: -\( |S|(|V| - |S|) \)

\[ v^T M v = v^T v - (2|E(S, S)||S|(|V| - |S|) \]

\[ \frac{v^T M v}{v^T v} = 1 - \frac{2|E(S, \overline{S})|}{|S|} \]

\[ \lambda_2 \geq 1 - 2h(S) \rightarrow h(G) \geq \frac{1 - \lambda_2}{2} \]
See you ...

Thursday.