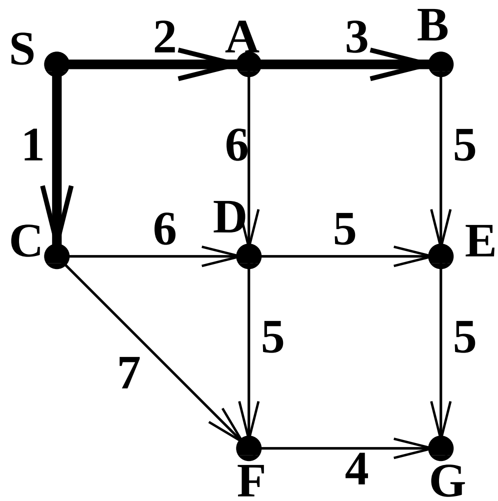
1. CLR 25.3-1
2. CLR 25.3-4. Modify Bellman-Ford so that it sets $d[v] = -\infty$ for all vertices v for which there is a negative weight cycle on some path from the source to v.
3. CLR 24-1. *Second-best* minimum spanning tree.
4. Suppose that, after finding the minimum spanning tree of a weighted graph, you are told that the length of an edge is *increased*. You are given the edge, and the amount by which its length is *increased*. Give an algorithm for determining the new minimum spanning tree. Your algorithm should be as fast as possible. (An obvious solution is to run the MST algorithm again. Your answer should be faster than that.)

   Answer the above question again with *decreased* instead of *increased*.
5. CLR 24.2-6.
6. *Sample question from from the Fall 1997 Midterm.* Give a linear-time algorithm which, given a connected undirected graph, returns a node whose deletion would *not* disconnect the graph. Give
   - Brief description or pseudocode
   - Justification of correctness
   - Running time and justification
7. *Sample question from from the Fall 1997 Midterm.* **True or False?** *For this homework, a full explanation is required for each part.*
   - Two directed graphs G_1 and G_2 have the same number of nodes, G_1 is a DAG, while G_2 is not a DAG. Then G_2 must have fewer strongly connected components than G_1.
   - If we add a directed edge to a directed graph, the number of strongly connected components cannot increase.
   - Adding a directed edge to a directed graph may decrease the number of strongly connected components by at most one.
   - If all edge weights are distinct, the minimum spanning tree is unique.
   - If all edge weights are distinct, the *second best* spanning tree is unique.
   - If all edge weights are distinct, the *maximum* spanning tree is unique.
   - In a d-heap, `deletemin` is cheaper than `insert.`
   - Consider a weighted undirected graph, a cycle in it, and an edge in the cycle whose weight is larger than the weight of any other edge in this cycle. Then this edge will never be a part of the minimum spanning tree.
   - Consider a weighted undirected graph, a cycle in it, and an edge in the cycle whose weight is smaller than the weight of any other edge in this cycle. Then this edge will always be a part of the minimum spanning tree.

8. *Sample question from from the Spring 1998 Midterm.*

We are running one of these three algorithms on the graph below, where the algorithm has already "processed" the bold-face edges. (Ignore the directions on the edges for Prim's and Kruskal's algorithms.)

- Prim's for the minimum spanning tree, starting from s.
- Kruskal's for the minimum spanning tree.
- Dijkstra's for shortest paths from s.



(a) Which edge would be added next in Prim's algorithm?

(b) Which edge would be added next in Kruskal's algorithm?

(c) Which vertex would be marked next in Dijkstra's algorithm, i.e. deleted from the top of the heap? Which final edge would Dijkstra's algorithm choose as part of the shortest path to this vertex (i.e. which edge connects to this vertex as part of the shortest path from s)?