CS270 Problem Set 2 — Due: Thursday April 20.

Rules: You can work in groups and consult papers and books. But you should write your paper by yourself, and you should give credit in it where credit is due, sources and colleagues. Please also write briefly, neatly, and correctly.

1. Project. Write a 300-word executive summary of your project: problem definition, background, previous work, approach, desired results, and progress made so far.

2. Dynamic Programming (or greedy)

   (a) Give an $O(n \log n)$ time algorithm for the Carpenter's problem.

   (b) Describe as efficient as you can algorithm that given a set of n points on the plane, and a number k, finds the least bumpy left to right path that uses exactly k points. (Bumpy is defined as the sum of the absolute values of the angle changes on the path.)

   (c) Describe as efficient as you can algorithm that given a set of n numbers, $a_1, \ldots a_n$, and a number k, that determines if a subset of the numbers sums to k. Describe how your algorithm could output the set of numbers. Give the runnning time of your algorithm.

   (d) Describe an algorithm, that given a graph with edge weights and costs, and two nodes s and t, finds the cheapest path from s to t of weight at most W. (Your running time may be linear in W, and you can assume that every edge has weight at least 1, if you like.)

3. Coding.

   (a) Show that $\log \binom{n}{p}$ is appropriximately $\log H(p)n$ for large n.

   (b) In terms of $H(p)$, about how many strings have at most p fraction of ones?

   (c) Recall that Reed Solomon codes used polynomials over a field to encode messages. Assume that one wishes to send n bits of information in a k-bit codeword, what is the maximal number of bit errors that can be tolerated using Reed Solomon.

   (d) The answer to the previous question should not have been better than $\Theta(k/\log n)$. Recall that random linear codes have quite high minimum distance. Unfortunately, decoding a random linear code can be difficult. Show how to combine this result on random linear codes along with Reed Solomon codes to obtain a code that can tolerate $\Omega(k)$ errors. (That is, tolerate channels with a constant error rate.)

   (e) Recall the error correcting codes described in class that were built from expander graphs. These are described in the last section of
   "http://www.cs.yale.edu/homes/spielman/AEC/lect14.ps."
   Are these codes tolerant to $\Omega(k)$ errors? (Hint: this is not hard.)

4. **[12 pts] Integer Programming.** An integer program is a linear program in which the solutions are required to be integers. Integer programming is NP-hard, so any problem in NP can be reduced to it. Show how to write the following problems as integer programs.

   (a) MAX3SAT: Given a 3CNF formula, find an assignment which maximizes the number of satisfied clauses.

(b) TSP: Given a weighted, undirected graph, find a minimum-weight tour of the graph — a cycle which visits all nodes exactly once.

(c) VERTEX COVER: Given an undirected graph $G = (V, E)$ and an integer k, select a set $S \subseteq V$ of k nodes which maximizes the number of edges "touching" S, i.e. the number of edges (v, w) for which $v \in S$ or $w \in S$.

(d) 3D MATCHING: Given three disjoint sets W, X, and Y of n items, and a set M such that each element of M is a triple $(w, x, y) \in W \times X \times Y$, find a *maximum matching*: a largest subset of S of M such that each $w \in W$, each $x \in X$, and each $y \in Y$ appears at most once among all triples in S. For example, if $n = 3$ and $M = \{(x_1, y_1, z_1), (x_1, y_1, z_2), (x_2, y_3, z_1), (x_3, y_2, z_3)\}$, then the last three triples in M form a maximum matching (in this case we know it is maximum since no matching can have size greater than n).

(e) 2-PROCESSOR SCHEDULING: Given a list of integer job lengths $\ell_1, \ldots, \ell_n$, find a *schedule* of the jobs on two processors with minimum *makespan*. A schedule assigns each job either to processor 0 or processor 1, and the makespan of a schedule is the total length of jobs on the most-loaded of the two processors.

(f) DEGREE CONSTRAINED SPANNING TREE: Find a spanning tree of a given graph such that the maximum degree of a vertex in the tree is minimized.

5. NP-completeness: The problem of finding an assignment a CNF formula with two variables per clause that maximizes the number of satisfied clauses is the MAX 2-SAT problem. Show that this problem is NP-complete.

6. Approximation Algorithms.

   (a) Consider the minimum delay set cover problem, where one is given m elements and n sets, and the cover consists of an ordering of the sets. The cost of covering an element is the number of the first set in this ordering that covers it. For example, if an element is in the first set it has cost 1. The problem is to find the ordering that minimizes the total cost of covering all the elements. Give an approximation algorithm for solving this problem with as good an approximation factor as you can.

   (b) Consider a set covering problem where each element is in at most 3 sets. Give a factor of three approximation algorithm for solving this problem.

   (c) Give a factor of three approximation algorithm for 3 dimensional matching.

   (d) Give a better approximation algorithm for 3 dimensional matching, if you can.

   (e) Give an example class of graphs where the MST of a graph in your class is approximately a factor of two cheaper than the TSP of the graph (assuming triangle inequality.) Describe a solution that Christofides algorithm may come up with on your graph.