CSE 521: Design and Analysis of Algorithms Homework #3
Some random problems; some linear programming. April 23rd, 2009.

Due: May 7th, 2009. Reading: Lecture notes, Chapter 7 of DPV book.

---

1. **Selection.** [15 points]

   QuickSelect is the following simple randomized algorithm for finding the kth smallest element in an unsorted set S.

   QuickSelect(S, k):

   1. Pick a pivot element p uniformly at random from S.
   2. By comparing p to each element of S, split S into two pieces: $S_1 = \{x \in S : x < p\}$ and $S_2 = \{x \in S : x > p\}$.
   3. If $|S_1| = k - 1$, then output p.
      If $|S_1| > k - 1$ then output QuickSelect(S_1, k).
      If $|S_1| < k - 1$ then output QuickSelect$(S_2, k - |S_1| - 1)$.

   Prove that the expected number of comparisons made by QuickSelect on a set S of size n is at most $3.5n$. (Partial credit for proving it's at most $4n$.)

2. **Approximate min cuts.** [15 points]

   In this problem, we revisit the Contraction algorithm for computing minimum cuts, and consider its ability to find near-minimum cuts. For an integer $\ell \geq 1$, define an ℓ-approximate cut to be a cut whose size is at most ℓ times the size of the minimum cut. (We are considering unweighted, undirected graphs in this problem).

   (a) Prove that a single trial of the contraction algorithm yields as output an ℓ-approximate cut with probability at least $\Omega(n^{-2/\ell})$, where n is the number of vertices in the graph.

   (b) For each *fixed* integer $\ell \geq 1$, give a polynomial time algorithm that outputs a list of all ℓ-approximate cuts in the graph. Prove also that, in any n-vertex graph, there are at most $n^{2\ell}$ ℓ-approximate cuts.

3. **A simplex example.** [15 points]

   (a) Solve the following linear program using the simplex method. Start with the initial solution $(0, 0, 1)$. Show your intermediate steps.

   Maximize $3x_2 + x_3$ subject to

$$\begin{aligned} x_1 + 2x_2 + x_3 &\leq 2 \\ 2x_1 + x_2 - x_3 &\leq -1 \\ 3x_1 + 2x_2 + x_3 &\leq 3 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

(b) [Removed]

4. **Integrality and matchings.** [25 points]

(a) In this problem, you will identify a very useful criterion for a linear program to have an integral optimum solution. Consider an LP in standard form:
Maximize $c \cdot x$ subject to

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

where A is an integer matrix and b is an integer vector, but the vector c need not be integral. Suppose that the determinant of every square submatrix of A is 0,1, or -1. Prove that if the linear program is bounded and has an optimal solution, then it has an optimal solution x^* whose entries are integers.

Problem: Although you will need the full strength of the preceding assertion to finish (b)-(e), you are only required to prove that there is an integral optimal solution *when* A *is non-degenerate,* i.e. when any subset of d rows of A are linearly independent.

Hint: Use the fact that when x is a vertex of the feasible region, we have $A_T x = b_T$ where A_T is a $d \times d$ sub-matrix of A, and b_T is a subset of d coordinates from b. (Recall that we used this fact in the proof of strong duality.)

(b) Consider the problem of computing a maximum matching in a bipartite graph. Express this as an integer linear program using a $0/1$-valued variable x_e to indicate whether an edge e is picked in the matching or not.

(c) Consider the linear programming relaxation of the above integer linear program, obtained by relaxing the constraint $x_e \in \{0, 1\}$ to $x_e \geq 0$. Write down the dual linear program of this LP relaxation. What optimization problem is the dual a natural relaxation of?

(d) Prove, using part (a), that the linear program above for maximum matching on bipartite graphs as well as its dual both have integral optimum solutions.

(e) What "min-max theorem" concerning bipartite graphs can you conclude using part (d)?