EE 360C — Algorithms — Summer 2013
Homework #1

Due: June 26, 2013 11:30am (in class)

Homework problems are to be done individually. You may discuss the problem and general concepts with other students, but you must write your solutions independently.

Each question is worth 10 points. Maximum possible score is 30.

---

1. Consider the day care center problem described in your Programming Assignment #1.

   (a) Explain the Gale and Shapely algorithm for finding an assignment of kids to nannies such that it is kid-optimal, i.e., the kids get assigned to their best possible nannies.

   (b) Prove that any kid that is chosen by the kid-optimal algorithm is also chosen by *some* nanny in the nanny-optimal algorithm.

   (c) Show that there is *no* matching, stable or not, which is preferred by *all* kids over the kid-optimal matching.

2. Suppose you have algorithms with the six running times listed below. (Assume these are the exact number of operations performed as a function of the input size n.) Suppose you have a computer that can perform 10^{10} operations per second, and you need to compute a result in at most an hour of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour?

   (a) n^2 (b) n^3 (c) $100n^2$ (d) $n \log n$ (e) 2^n (f) 2^{2^n}

3. There's a natural intuition that two nodes that are far apart in a communication network—separated by many hops—have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here's one that involves the susceptibility of paths to the deletion of nodes.

   Suppose that an n-node undirected graph $G = (V, E)$ contains two nodes s and t such that the distance between s and t is strictly greater than $n/2$. Show that there must exist some node v, not equal to either s or t, such that deleting v from G destroys all $s - t$ paths. (In other words, the graph obtained from G by deleting v contains no path from s to t.) Give an algorithm with running time $O(m + n)$ to find such a node v.