600.363 Introduction to Algorithms / 600.463 Algorithms I **Fall 2014**
Homework #9 **Due:** December 2, 2014, 1:30pm

Please start each problem on a new page, and include your name on each problem. You can submit on blackboard, under student assessment.

Remember: you may work in groups of up to three people, but must write up your solution entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. You may use the internet to look up formulas, definitions, etc., but may not simply look up the answers online.

Please include proofs with all of your answers, unless stated otherwise.

1 Bin Packing (33 points)

Suppose that we are given a set of n objects, where the size s_i of the ith object satisfies $0 < s_i < 1$. We wish to pack all the objects into the minimum number of unit-size bins. Each bin can hold any subset of the items whose total size does not exceed 1. Let $S = \sum_{i=1}^{n} s_i$.

(a) Prove that the optimal number of bins required is at least $\lceil S \rceil$.

The *first-fit algorithm* considers each object in turn (from 1 to n) and places it in the first bin that can accommodate it. If there is no such bin, then we create a new bin for it and make it the last bin. Note that this defines an ordering over bins based on when we created them, so "first" and "last" make sense.

(b) Prove that the first-fit algorithm leaves at most 1 bin at most half full. In other words, all bins but 1 are more than half full.

(c) Prove that the first-fit algorithm is a 2-approximation to bin packing.

2 TSP (33 points)

Given a weighted undirected graph G, a traveling salesman tour for G is the shortest tour that starts at some node, visits all the vertices of G, and then returns to the start. We will allow the tour to visit vertices multiple times (so, our goal is the shortest cycle, not the shortest simple cycle). This version of the TSP that allows vertices to be visited multiple times is sometimes called the *metric* TSP problem, because we can think of there being an implicit complete graph H defined over the nodes of G, where the length of edge (u, v) in H is the length of the shortest path between u and v in G. (By construction, edge lengths in H satisfy the triangle inequality, so H is a metric. We?re assuming that all edge weights in G are positive.)

(a) Briefly: show why we can get a 2-approximation to the TSP by finding a minimum spanning tree T for H and then performing a depth-first traversal of T. (If you get stuck, this is done in much more detail than necessary in CLRS 35.2.1.)

(b) The MST T must have an even number of nodes of odd degree. In fact, *any* undirected graph must have an even number of nodes of odd degree. Prove this.

(c) Let M be a minimum-cost perfect matching (in H) between the nodes of odd degree in T. So if there are $2k$ nodes of odd degree in T, then M will consist of k edges between these nodes, none of which share an endpoint. Prove that the total length of edges in M is at most one-half the length of the optimal TSP tour.[1]

An *Euler tour* of a graph is a cycle that traverses every edge exactly once (so it may visit the same node multiple times, and so is not a *simply* cycle). It turns out that a graph has an Euler tour if and only if every vertex has even degree. In fact, this is true even of multigraphs (graphs in which the same edge can appear multiple times).

(d) Combine the above fact about Euler tours and the previous parts of this problem to get a 3/2-approximation for TSP. Hint: consider the (multi)graph obtained by taking the union of T and M.

This algorithm is due to Christofides [1976]. It is still the best known algorithm for metric TSP.

3 Representatives (33 points)

One technique sometimes used in the sciences is to build a representative set of data and study it intensively, and use it to learn (by inference) properties about the full data set. For example, a well-known protein chemist has proposed studying a small set of proteins very intensively, rather than having many research groups studying many different proteins. In order to be useful, a representative set of proteins should clearly have two properties: 1) it should be relatively small, and 2) every protein in the full collection should be "similar" to some protein in the representative set.

To formalize this, we suppose that there is a large set P of proteins, and between any two proteins p and q there is some distance $d(p, q)$ which measures how similar they are (in fact, scientists typically use some notion of sequence alignment distance, which we briefly talked about when we were studying dynamic programming). We are also given some cutoff Δ, and we say that p and q are *similar* if $d(p, q) \leq \Delta$. We say that a subset Q of P is a *representative set* if for every protein $p \in P$, there is some protein $q \in Q$ such that p and q are similar (i.e., $d(p, q) \leq \Delta$).

Our goal is to compute a representative set of minimum size. Give an $O(\log n)$-approximation for this problem.

[1] In class we showed how to do min-cost perfect matching when the graph is bipartite (see Lecture 19), but it turns out that there are also efficient algorithms for min-cost perfect matching in arbitrary graphs.