Applied Algorithm Design: Exam

Prof. Pietro Michiardi

Rules and suggestions

- The idea is to complete first the questions and then the exercises.
- Having **the good** answer for all questions will guarantee to pass the exam with 10/20.
- Exercises are difficult if you are not familiar with the course contents. Suggestion: pick one exercise that you think is more feasible for you and focus on that one.

Questions

1. Discuss the data structures that must be used in the implementation of the Gale-Shapley algorithm.

2. When does a bipartite graph have a perfect matching?

3. Give an example of a "divide-and-conquer" algorithm.

4. Give at least two properties of a tree graph.

5. In a content resolution protocol, you need to set the probability p for a process to access a shared resource that can be used by one and only one process at a time. Give an intuition on how to set p. Optionally, show mathematically what is the best p value.

6. Given a digraph $G = (V, E)$ and two nodes s and t, find the minimum number of edges whose removal disconnects t from s.
   (a) Given this problem, which algorithm can be used to find the solution?
   (b) Which is the running time of this algorithm?
   (c) Why is this question is important in real systems?

Exercise 1

Consider a town with n men and n women seeking to get married to one another. Each man has a preference list that ranks all the women, and each woman has a preference list that ranks all the man.

The set of all $2n$ people is divided into two categories: *good* and *bad* people. Suppose that for some number k, $1 \leq k \leq n-1$, there are k good men and k good women; thus there are $n-k$ bad man and $n-k$ bad women.

Everyone would rather marry any good person than any bad person. Formally, each preference list has the property that it ranks each good person of the opposite gender higher than each bad person of the opposite gender: its first k entries are the good people (of the opposite gender) in some order, and its next $n-k$ are the bad people (of the opposite gender) in some order.

Question: Show that in every stable matching, every good man is married to a good woman.

Exercise 2

The Center Selection Problem we have seen in class has been studied through the lenses of approximation algorithms.

Here, instead, we take a simple local search approach to the problem. In this problem, we are given a set of sites $S = \{s_1, s_2, ..., s_n\}$ in the plane, and we want to choose a set of k centers $C = \{c_1, c_2, ..., c_k\}$ whose covering radius[1] is as small as possible.

We start arbitrarily choosing k points in the plane to be the centers $c_1, c_2, ..., c_k$. We then alternate the following two steps.

1. Given the set of k centers $c_1, c_2, ..., c_k$ we divide S into k sets: For $i = 1, 2, ..., k$ we define S_i to be the set of all the sites for which c_i is the closest center
2. Given this division of S into k sets, construct new centers that will be as "central" as possible relative to them. For each set S_i, we find the smallest circle in the

[1] The farthest that people in any one site must travel to their nearest center.

plane that contains all points in S_i, and define the center c_i to be the center of this circle.

If steps 1) and 2) cause the covering radius to strictly decrease, then we perform another iteration; otherwise the algorithm stops.

The alternation of steps 1) and 2) is based on the following natural interplay between sites and centers. In step 1) we partition the sites as well as possible given the centers; and then in step 2) we place the centers as well as possible given the partition of the sites.

Question 1: Prove that this local search algorithms eventually terminates.

Question 2: Consider the following statement.

There is an absolute constant $b > 1$ (independent of the input of the algorithm), so when the local search algorithm terminates, the covering radius of its solution is at most b times the optimal covering radius.

Decide whether you think this statement is true or false, and give a proof of either the statement or its negation.

[Recall]: **The Center Selection Problem**

- We have a set $S = \{s_1, s_2, \cdots, s_n\}$ of n sites to serve
- We have a set $C = \{c_1, c_2, \cdots, c_k\}$ of k centers to place

The problem: Select k centers C placement so that maximum distance form a site to the nearest center is minimized

Exercise 3

Recall the *Shortest-First* greedy algorithm for the Interval Scheduling Problem: Given a set of intervals, we repeatedly pick the shortest interval I, delete all the other intervals I' that intersect I, and iterate.

During class we saw that this algorithm does *not always* produce a maximum-size set of non-overlapping intervals. However, it turns out to have to following interesting *approximation guarantee*. if s^* is the maximum size of a set of non-overlapping intervals, and s is the size of the set produced by the *Shortest-First* algorithm, then $s \geq \frac{1}{2}s^*$, that is the *Shortest-First* algorithm is a 2-approximation.

Question: prove this fact.

[Recall] **The Interval scheduling problem**:
We have a set of requests $\{1, ..., n\}$ where the i^{th} request corresponds to an interval

of time starting at $s(i)$ and finishing at $f(i)$. We say that a subset of the requests is *compatible* if no two of them overlap in time.

Our goal is to accept as large a compatible subset as possible. Compatible sets of maximum size will be called *optimal*.

Exercise 4

Suppose we have a collection of small, low-powered devices scattered around a building. The devices can exchange data over short distances by wireless communication, and we suppose for simplicity that each device has enough range to communicate with d other devices.

Thus we can model the wireless connections among these devices as an undirected graph $G = (V, E)$ in which each node is incident to exactly d edges.

Now, we'd like to give some of the nodes a stronger *uplink transmitter* that they can use to send data back to a base station. Giving such a transmitter to *every node* would ensure that they can all send data with this stronger device, but we can achieve the same result (all node can send data to the base station) while handing out fewer of this strong transmitters.

Suppose that we find a subset S of the nodes with the property that every node in $V \setminus S$ is adjacent to a node in S. We call such a set S a **dominating set**, since it "dominates" all other nodes in the graph. If we give uplink transmitters only to the nodes in a dominating set S, we can still extract data from all nodes: Any node $u \notin S$ can choose a neighbor $v \in S$, send its data to v, and have v relay the data back to the base station.

The issue is now to find a dominating set S of minimum possible size, since this will minimize the number of uplink transmitter we need. This is a NP-hard problem.

Despite the NP-hardness, it's important in application like this to fina as small a dominating set as one can, even if it is not optimal. We will see here that a simple *randomized strategy* can be quite effective. Recall that in our graph G, each node is incident to exactly d edges. So clearly any dominating set will need to have size at least $\frac{n}{d+1}$, since each node we place in a dominating set can take care only of itself and its d neighbors. We want to show that a random selection of nodes will, in fact, get us quite close to this simple lower bound.

Question: show that for some constant c, a set of $\frac{cn \log n}{d+1}$ nodes chosen uniformly at random from G will be a dominating set with high probability.

`Hint`: In other words, this completely random set is likely to form a dominating set that is only $O(\log n)$ times larger than our simple lower bound of $\frac{n}{d+1}$.