CS4234: Optimization Algorithms

Problem Set 4

Due: September 8, 11:59am

Instructions. Solve all the S-problems (standard). This week there are no A-problems this week. (Review the A-problems from previous weeks, if you like.) This problem set includes one problem related to Eulerian Cycles, and one problem related to a new combinatorial optimization problem that we have not yet seen in class.

Exercises (not to be submitted)

Ex-1. Give an example where the 2-approximation algorithm for the travelling salesman problem is not optimal (even when the distances are metric, or even Euclidean).

Ex-2. Give an algorithm for solving the Travelling Salesman Problem on n nodes in $O(2^n n^3)$ time. (Notice that there are 2^n different subsets of n. Use dynamic programming.)

Standard Problems (to be submitted)

S-1. Humperdink and Mildred run two hotels in a popular tourist destination (see Figure 1). Tourists, when they visit town, often want to see everything: they wake up in the morning, leave their hotel, attempt to see all the sights, and end the day at the Central Plaza (where there is an excellent restaurant for dinner). Unfortunately, as things stand, tourists staying at both hotels cannot accomplish this without retracing their steps at some point during the day.

Problem 1.a. To solve this problem, Humperdink decides to build a new road. Which road should he build so that his guests can start at his hotel, traverse every road exactly once, and end at the Central Plaza? Draw this new route on the graph.

Problem 1.b. Mildred is upset that Humperdink has gained an advantage. What road should Mildred build to ensure that her guests can all perform a good tour (starting at her hotel, traversing every road, and ending at the Central Plaza), but none of Humperdink's guests can perform a good tour?

Problem 1.c. Recall that an *Eulerian Cycle* for a multigraph G is a cycle that crosses each edge exactly once. Given a special starting node s and a destination d, we define an *Eulerian Path* as a path that starts at s, ends at d, and crosses each edge exactly once. Prove the following statement: *A connected multigraph G has an Eulerian Path if and only if s and d have odd degree, and every other node has even degree.*

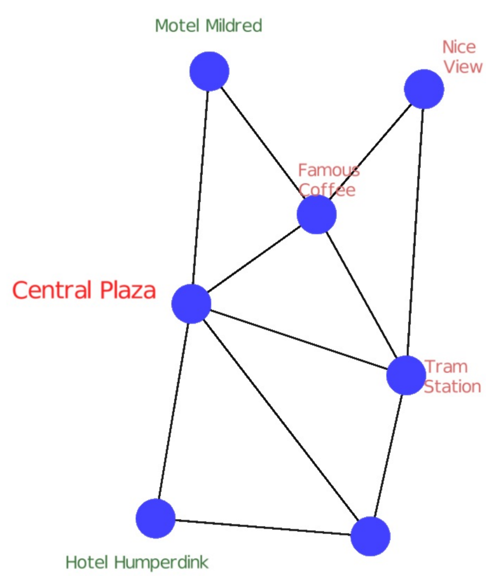

Figure 1: Map of a popular tourist destination, including Hotel Humperdink, Motel Mildred, and the Central Plaza (along with assorted other locations).

S-2. A new museum has just opened in Singapore, and everyone wants to go. Alas, they can only let in n people per day. They have hired you as a consultant to help maximize their attendance.

They have the following problem: people arrive at the museum in groups. Sometimes in small groups (e.g., a family) and sometimes a large group (eg., a tour group, or a school class). When a group wants to visit the museum, there must be space for all of the group to enter the museum. If there isn't room for the entire group, then the entire group is turned away. The museum wants your advice on how to let as many people visit as possible.

You are given a set of groups $G[1\,..\,m] = [g_1, g_2, \ldots, g_m]$ where g_i is a number representing the size of the group. Each of these groups is trying to make a reservation to enter the museum next week. Assume that you can admit at most n people total.

Problem 2.a. Consider the following greedy admission algorithm, where the function Admit(i) admits group i and Reject(i) rejects group i.

```
Algorithm: GroupAdmission(G[1..m], n)
Procedure:
    admitted ← 0
    remaining ← n
    G ← Sort(G)
    for i ← 1 to m do
        if G[i] ≤ remaining then
            Admit(i)
            remaining ← remaining − G[i]
            admitted ← admitted + G[i]
        else
            Reject(i)
    return admitted
```

The greedy algorithm sorts the groups by size, with the largest groups first. It then iterates through the groups from largest to smallest, admitting any group for which there is room. Show that this algorithm works pretty well, i.e., show it is a 2-approximation of optimal. (That is, if, given G and n, it is possible to admit k people, then the greedy admission algorithm will admit at least $k/2$ people.)

Problem 2.b. Unfortunately, the GroupAdmission algorithm does not work perfectly. Show that the algorithm is not optimal by giving a counterexample in which, asymptotically as n gets large, the ratio between greedy seating and optimal seating approaches $1/2$.

Problem 2.c. Assume that the groups G are provided in read-only memory. That is, you cannot sort the groups G, nor can you copy the array. (You may only use $O(1)$ extra space.) In this context, give an efficient algorithm, and prove that it is a correct 2-approximation algorithm. (*Hint:* one solution runs in time $O(m \log n)$; however there is a better solution that runs in time $O(m)$.)

S-3. Review the exercises from Weeks 1–3. Review the lectures from Weeks 1–4. Come up with questions that we should answer in tutorial next week. Ask at least one question on the IVLE Survey for Week 5.