15-451/651 Algorithms, Spring 2018

Homework #4 **Due: March 6–8, 2018**

This is an oral presentation assignment. Same rules as HW2, and as for previous bonuses. Prove your answers correct for problems 1-3.

(25 pts) 1. **(Coffee, Anyone?)** The coffee chain ⋆⋆⋆\$\$\$\$ asks you to locate their $s \geq 1$ shops on the PA turnpike. You model the turnpike as a straight line. There are $n \geq s$ cities on the turnpike, which are located at positions $0 = a_1 < a_2 < \ldots < a_n$ on the line, with distances on the line modeling driving distances between the cities. You are allowed to locate the shops anywhere on the line, even on locations that are not cities. If you decide to locate the s shops at positions $b_1, b_2, \ldots, b_s$, the "pain" of this solution is the distance from an average city to its closest shop, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n}\left(\min_{j=1}^{s}|a_i - b_j|\right).$$

Give a dynamic programming algorithm to find the solution (i.e., the location of shops) that incurs the least pain, and that runs in time $O(n^3)$. It should not depend on the values of a_i.

Hint: consider subproblems that correspond to a prefix $\{a_1, \ldots, a_i\}$ of the cities and a number of shops $s' \leq s$. One more useful fact is that we need only consider solutions that place shops at some subset of the cities $\{a_i\}$ (i.e., $\{b_j\} \subseteq \{a_i\}$); this can be seen most easily in the case of $s = 1$ where if the shop is between a_i and a_{i+1}, we can simply move it in whichever direction has more cities, or to either one if there is a tie. You may use this fact without proof, but convince yourself that it is true.

(25 pts) 2. **(ClubTown.)** The n residents of ClubTown like belonging to clubs, but no one wants to be the chairperson of these clubs. Say there are m clubs, and S_i is the set of people in club i. For each club i, one of the people $j \in S_i$ must be the chairperson. Since people would rather not be chair, they want the work of being chairperson to be divided as fairly as possible. Your task in this problem is to give an algorithm to do this efficiently.

The fairness criterion is the following: Say that person p is in some k of the sets, which have sizes $n_1, n_2, \ldots, n_k$, respectively. Person p should really have to be the chair for $\frac{1}{n_1} + \frac{1}{n_2} + \cdots + \frac{1}{n_k}$ clubs, because this is the amount of "club-ness" resource that this person effectively uses. Of course this number may not be an integer, so let's round it up to an integer. This quantity $\lceil \frac{1}{n_1} + \frac{1}{n_2} + \cdots + \frac{1}{n_k} \rceil$ is called their *fair cost.* A *fair solution* is an assignment of chairpersons to clubs so that that each person is chair for no more clubs than their fair cost.

> Say the first club has Alice and Can, and the second has Alice, Tanvi, and Shalom. Alice's fair cost would be $\lceil 1/2 + 1/3 \rceil = 1$. So Alice chairing both clubs would not be fair. Any solution except that one is fair.

Note that it is not clear there even *exists* a fair solution. Give a polynomial-time algorithm that, given $S_1, S_2, \ldots, S_m$, always finds a fair solution. (This will also show that there always exists a fair solution.)

(25 pts) 3. **(Rental Carts.)** You start a new cart rental company Hürtz. You will locate cart rental offices in some subset S of n cities, to produce the most profit. Profit is the revenue minus cost.

The ith cart rental office will cost $a_i \geq 0$ dollars to build, so the total cost of building offices at some subset S of the n cities is $\sum_{i \in S} a_i$.

You get revenue from these rental offices too. People drive carts only between distinct cities. There are m of the $\binom{n}{2}$ pairs of cities such that people drive carts between them, and you can get some revenue for that. So, as input, we are also given m triples (x_i, y_i, r_i), such that the company gets revenue of $r_i \geq 0$ dollars if it builds on both locations x_i and y_i.

> E.g., if the costs were $7, 9, 12, 14$ and the triples were $\{(1,2,14), (1,3,11), (2,3,22), (1,4,6)\}$, then building an office at just location 1 gets profit -7, building at $\{1,2\}$ gets profit $14 - (7+9) = -2$, building at $\{1,2,3\}$ gets profit $(14+11+22) - (7+9+12) = 19$, etc.

Give an efficient algorithm to find a subset S of these n locations the company should build in order to earn the most profit. (Hint: Use maximum s-t flow.)

(25 pts) 4. **(Delightful Dominos.)** In this programming problem you're given a rectangular board of square cells. Some of the cells are blocked. You are to compute the maximum number of dominos (1×2 tiles) that can be placed on the board. Each domino will be in either a vertical or horizontal orientation, and is placed on two neighboring squares, neither of which must be blocked. The time limit is 10 seconds.

Input: The first line contains two space-separated integers: r and c. The next r lines are strings of length c comprised of the characters "`.`" and "`x`". The "`x`" characters denote cells of the board that are blocked. $1 \leq r, c \leq 30$.

Output: Print the maximum number of dominos that can be placed on the board satisfying these constraints. The format of the output is illustrated below.

Input 1:

```
2 3
...
...
```

Output 1:

```
3 dominos
```

Input 2:

```
5 4
.xx.
xx.x
x...
xx.x
.xx.
```

Output 2:

```
1 domino
```

Input 3:

```
2 5
..xx.
.....
```

Output 3:

```
4 dominos
```