

# CS 473: Algorithms, Spring 2018

## HW 2 (due Wednesday, February 7th at 8pm)

This homework contains three problems. **Read the instructions for submitting homework on the course webpage.**

**Collaboration Policy:** For this home work, each student can work in a group with up to three members. Only one solution for each group needs to be submitted. Follow the submission instructions carefully.

1. The beautiful city of Malsgorith has a city center (CC), and the broad avenues of the city form a tree pattern that begins from the city center (considered here to be the root of the tree pattern). The nodes of this tree pattern are considered to be intersections of the avenues, and there are  $n$  of them. Each avenue has length of one mile. Furthermore, in this tree pattern if there exists an avenue from intersection  $u$  to intersection  $v$  and another avenue from intersection  $v$  to intersection  $w$ , in other words, ( $\#$  miles from  $w$  to CC) =  $1 + (\#$  miles from  $v$  to CC) =  $2 + (\#$  miles from  $u$  to CC), then there exist another smaller road from  $u$  to  $w$ . Essentially the entire network is as follows:

- (a) Avenues form tree  $T$  rooted at CC.
- (b) In addition, every node in  $T$  is connected to its grandchildren in  $T$  by a smaller road.

It is your first day as the city's electrical expert and you are tasked with providing lighting to all the avenues and roads. Lights can only be installed at the intersections, and a light source at intersection  $u$  has an installation cost  $c(u) > 0$  and lights all the avenues and roads that are incident at  $u$ . Being an expert in your field, design an algorithm to compute the cost of the minimum cost installation.

2. You are given a sequence of numbers  $a_1, a_2, \dots, a_n$ . You are playing a game against an opponent, where you both take turns to pick a number from either end of the sequence. The game ends when no number remains to be picked, and the one with the largest sum of numbers picked wins. For example, if the sequence is 3, 6, 1, 9, 5 and it is your turn to play, then you can either choose 3 or 5, but no other intermediate number. Suppose you pick 5, then it is your opponent's turn, the sequence become 3, 6, 1, 9, and they can choose either 3 or 9.

Suppose you always start first, and your opponent always greedily picks the highest of the two available numbers.

- (a) Prove that you should not also use the greedy strategy. That is, show that there is a game that you can win, but only if you do not follow the same greedy strategy.
  - (b) Describe and analyze an algorithm to determine the maximum total number that you can collect playing against the greedy opponent.
3. Consider the following bin packing problem. The input consists of  $n$  items with sizes  $w_1, w_2, \dots, w_n$  and that needs to be packed into bins  $B_1, B_2, \dots, B_m$  of large enough sizes. The

amount of a bin used is the sum of the sizes of the items packed in that bin. Given a packing of all the items into bins the *maximum load* is the maximum amount of any bin used. To find a packing that minimizes the maximum load is a well-known NP-hard problem.

- (a) Consider the setting where there are only 3 distinct item sizes  $\{s_1, s_2, s_3\}$ . That is,  $w_i \in \{s_1, s_2, s_3\}$  for  $1 \leq i \leq n$ . Describe a polynomial-time algorithm for this problem.
- (b) **[Extra Credit]** Consider now the setting where there are  $k$  distinct item sizes, where  $k$  is some fixed constant. Again, describe a polynomial-time algorithm for this problem.

**The remaining problems are for self study. Do *NOT* submit for grading.**

- Given an undirected graph  $G = (V, E)$  we defined its *square*, denoted by  $\text{square}(G)$  as the graph  $G' = (V, E')$  where  $uv \in E'$  iff there is a path of length at most 2 between  $u$  and  $v$  in  $G$ . That is,  $uv \in E'$  if  $uv \in E$  or if there is a node  $w$  such that  $uw, wv \in E$ . In class we saw an algorithm for the maximum weight independent set problem in a tree  $T = (V, E)$ . Design and analyze an algorithm for the maximum weight independent set problem for the square of a given tree  $T = (V, E)$ .
- Given a graph  $G = (V, E)$  a matching is a set of edges  $E' \subseteq E$  such that no two edges in  $E'$  share an end point. Describe an efficient algorithm that given a tree  $T = (V, E)$  and non-negative weights  $w : E \rightarrow \mathbb{R}_+$  finds a maximum weight matching in  $T$ .
- See Homework 2 from Spring 2016 taught by Jeff Erickson. <https://courses.engr.illinois.edu/cs473/sp2016/hw/hw2.pdf>
- See Homeworks 2 and 3 from Fall 2015 taught by Sarel Har-Peled. All the problems are on dynamic programming. They are a bit challenging. [https://courses.engr.illinois.edu/cs473/fa2015/w/hw/hw\\_02.pdf](https://courses.engr.illinois.edu/cs473/fa2015/w/hw/hw_02.pdf) and [https://courses.engr.illinois.edu/cs473/fa2015/w/hw/hw\\_03.pdf](https://courses.engr.illinois.edu/cs473/fa2015/w/hw/hw_03.pdf).
- There are several nice shortest path problems in both text books and at the end of Jeff's notes on shortest paths. Check them out.
- Given a directed graph  $G = (V, E)$  with non-negative edge lengths  $\ell : E \rightarrow \mathbb{R}_+$ , describe an algorithm that finds the shortest cycle in  $G$  that contains a specific node  $s$ . Describe an algorithm to find the shortest cycle containing  $s$  with at most  $k$  edges. Describe an algorithm to find a cycle  $C$  with minimum *average* length where the average length of a cycle  $C$  is defined as  $\ell(C)/|C|$ .