

Homework #3 (due in class December 3, 2019)

**Instructions: Submit your solution 10 minutes before the class begins.**

**Collaboration policy: You can discuss the problems with other students, but you must write the final answers by yourself. Please specify all of your collaborators (names and student id's) for each problem. If you solve some problems by yourself, please also specify “no collaborators”.**

**Homework without collaborator specification will not be graded.**

1. Exercise 16.2-5 (page 428)

Describe an efficient algorithm that, given a set  $\{x_1, x_2, \dots, x_n\}$  of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct.

2. Exercise 16.3-3 (page 436)

What is an optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?

a: 1 b: 1 c: 2 d: 3 e: 5 f: 8 g: 13 h: 21

Can you generalize your answer to find the optimal code when the frequencies are the first  $n$  Fibonacci numbers?

3. Modified Problem 16-1 (pages 446-447)

Consider the problem of making change for  $n$  dollars using the fewest number of coins. Assume that each coin's value is an integer.

- Suppose that the available coins are in the denominations that are powers of  $c$ , i.e., the denominations are  $c^0, c^1, \dots, c^k$  for some integers  $c > 1$  and  $k \geq 1$ . Describe a greedy algorithm that solves the problem.
- Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include a one dollar so that there is a solution for every value of  $n$ .
- Give an  $O(nk)$ -time algorithm that makes change for any set of  $k$  different coin denominations, assuming that one of the coins is one dollar.

4. Two players called  $P1$  and  $P2$  are playing a card game. In the game, there are  $m$  lines of cards on the table. (Each line of cards may have different numbers of cards.) On each card, there is a positive integer score on it. Each card is faced up and the score on the card is visible.

The players take turns choosing one card. (Assume  $P1$  always play first.) In  $P1$ 's turn he takes a card from the left of any non-empty line, and in  $P2$ 's turn he takes a card from the right of any non-empty line. Players want to maximize the sum of scores on the cards he took. The game ends when all lines become empty. Figure 1 is an example when  $m$  is 3.

Please give an  $O(n + m \log m)$  algorithm to find out the sum of scores that  $P1$  gets if both players play optimally ( $n$  is the total number of cards in the game). Briefly justify that your algorithm is correct.

Line 1	5	9	6	4
Line 2	7	9	6	
Line 3	2	3	8	1

Figure 1: The example shows a case when  $m$  is three. Line 1, Line 2 and Line 3 have 4 cards, 3 cards and 4 cards respectively. In this case, if both players play optimally,  $P1$  will get all the cards with the black outer frame. So, the sum of scores that  $P1$  gets is 35.

- Given an array consists of  $n$  positive integers  $a_1, a_2, \dots, a_n$  with the condition  $a_i \leq a_{i+1} \leq 2 \cdot a_i$  for any positive integer  $i$  ( $0 < i < n$ ).

For each number in the array, it is allowed to remain unchanged or turn negative. The task is to add negative sign on some numbers in the array so that the summation of all values in the array meets the limit  $\text{abs}(\sum_{i=1}^n a_i) \leq a_1$ . Please design an algorithm with  $O(n)$ -time that solve the problem. Briefly justify that your algorithm is correct.

- Exercise 22.2-7 (page 602).

There are two types of professional wrestlers: “babyfaces” and “heels”. Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have  $n$  professional wrestlers and we have a list of  $r$  pairs of wrestlers for which there are rivalries. Give an  $O(n+r)$ -time algorithm that determines whether it is possible to designate some of the wrestlers as babyfaces and the remainder as heels such that each rivalry is between a babyface and a heel. If it is possible to perform such a designation, your algorithm should produce it.

- Exercise 22.3-12 (page 612).

Show that we can use a depth-first search of an undirected graph  $G$  to identify the connected components of  $G$ , and that the depth-first forest contains as many trees as  $G$  has connected

components. More precisely, show how to modify depth-first search so that it assigns to each vertex  $v$  an integer label  $v.cc$  between 1 and  $k$ , where  $k$  is the number of connected components of  $G$ , such that  $u.cc = v.cc$  if and only if  $u$  and  $v$  are in the same connected component.

8. Exercise 22.4-4 (page 615)

Prove or disprove: If a directed graph  $G$  contains cycles, then  $\text{TOPOLOGICAL-SORT}(G)$  produces a vertex ordering that minimizes the number of "bad" edges that are inconsistent with the ordering produced.

9. Given a directed graph  $G$  consisting of  $V$  vertices and  $E$  edges. In graph  $G$ , every node has an out-degree at least  $n$ . Please design an algorithm that finds a simple cycle of length of at least  $n + 1$  in  $G$  with  $O(E)$ -time complexity. Briefly justify that your algorithm is correct. (A *simple cycle* of length  $m$  is a sequence of nodes  $v_1, v_2, \dots, v_m, v_{m+1}$  in which the only repeated vertices are the first and last vertices, and for any integer  $i$  ( $1 \leq i \leq m$ ) there exist an edge incident from  $v_i$  to  $v_{i+1}$ .)

10. Given an undirected graph  $G$  with  $V$  vertices and a sequence  $S$  which represents the BFS traversal order of a graph. We assume in graph  $G$  there is exactly one simple path between any two vertices. Please write an algorithm to determine whether  $S$  is a valid BFS traversal sequence of  $G$  with time complexity  $O(n \log n)$ . Briefly justify that your algorithm is correct. (In the BFS algorithm, the traversal order of the vertices depends on the order of choosing edges, so there may be multiple valid BFS traversal sequence.)