Algorithms and Data Structures (IBC027)

January 16, 2019

You are allowed to answer in Dutch. Whenever an algorithm is required, it can be given in pseudocode or plain English (or Dutch), and its running time and correctness must **always** *be justified (even informally, but in a clear way!). For problems 2, 3, 4, 5 and 7.1, where an algorithm must be defined, 20% of the points is for the correctness argument, and 20% for the analysis of the running time. The grade equals the sum of the scores for the seven problems below (plus possibly a bonus score for the homework assignments). Success!*

1 Quiz (1 point)

Question 1 Which of the following is not $\mathcal{O}(n^2)$?

(A) $(15^{10}) * n + 12099$

(B) $n^{1.98}$

(C) $n^3/\sqrt{n}$

(D) $(2^{20}) * n$

Question 2 Is following statement true or false? If a DFS of a directed graph contains a back edge, any other DFS of the same graph will also contain at least one back edge.

(A) true

(B) false

Question 3 An undirected graph G has n nodes. Its adjacency matrix is given by an $n \times n$ matrix whose diagonal elements are 0's, and non-diagonal elements are 1's. Which one of the following is true?

(A) Graph G has no minimum spanning tree (MST)

(B) Graph G has a unique MST of cost $n - 1$

(C) Graph G has multiple distinct MSTs, each of cost $n - 1$

(D) Graph G has multiple spanning trees of different costs

Question 4 Consider the following algorithms:

1. Breadth First Search
2. Depth First Search
3. Prim's Minimum Spanning Tree
4. Kruskal' Minimum Spanning Tree

What are appropriate data structures to use in an implementation of these algorithms?

(A) 1) Stack 2) Queue 3) Priority Queue 4) Union Find

(B) 1) Queue 2) Stack 3) Priority Queue 4) Union Find

(C) 1) Stack 2) Queue 3) Union Find 4) Priority Queue

(D) 1) Priority Queue 2) Queue 3) Stack 4) Union Find

2 Divide and conquer (1.5 points)

You are given a one dimensional array that may contain both positive and negative integers. Describe an $\mathcal{O}(n \log n)$ divide and conquer algorithm to find the sum of the contiguous subarray of numbers which has the largest sum. For example, if the given array is $[-2, -5, \mathbf{6}, \mathbf{-2}, \mathbf{-3}, \mathbf{1}, \mathbf{5}, -6]$, then the maximum subarray sum is 7 (see highlighted elements). The sum of the empty subarray is defined to be 0.

3 Graph algorithms (1.5 points)

After a wine tasting event organized by Thalia in the Mercator building, you decide to take the bus home. Since you planned ahead, you have a schedule that lists the times and locations of every stop of every bus that you could possibly use. Unfortunately, no single bus visits both the bus stop near the Mercator building and your home; you must change buses at least once. There are exactly B different buses. Each bus starts at 12:00am (midnight), makes exactly N stops, and finally stops running at 11:59:59pm. You are in time to catch the first bus that leaves from the bus stop closest to the Mercator building. Buses always run exactly on schedule, and you have an accurate watch. Finally, you are far too tired to walk between bus stops.

1. Describe and analyze an algorithm to determine the sequence of bus rides that gets you home as early as possible. Your goal is to minimize your arrival time, not the time you spend traveling.
2. Oh, no! The wine tasting event was held on Halloween, and the streets are infested with zombies! The bus company doesn't have the funding to add additional buses or install zombie-proof bus stops, especially for only one night a year. Describe and analyze an algorithm to determine a sequence of bus rides that minimizes the total time you spend waiting at bus stops; you don't care how late you get home or how much time you spend on buses. (Assume you can wait inside the Mercator building until your first bus is just about to leave.)

4 Dynamic programming (1.5 points)

Imagine you place a knight chess piece on a phone dial pad. This chess piece moves in an uppercase "L" shape: two steps horizontally followed by one vertically, or one step horizontally then two vertically:

Suppose you dial keys on the keypad using only hops a knight can make. Every time the knight lands on a key, we dial that key and make another hop. The starting position counts as being dialed. Give an efficient dynamic programming algorithm for the following problem: How many distinct numbers can you dial in N hops from a particular starting position? Specify the recursion equations on which your algorithm is based.

5 Network flow (1.5 points)

You are organizing a dance event, to be held all day Friday, Saturday, and Sunday. Several 30-minute sets of music will be played during the event, and a large number of DJs have applied to perform. You need to hire DJs according to the following constraints:

- Exactly k sets of music must be played each day, and thus $3k$ sets altogether.
- Each set must be played by a single DJ in a consistent music genre (ambient, bubblegum, dubstep, horrorcore, K-pop, Kwaito, mariachi, straight-ahead jazz, trip-hop, Nashville country, parapara, ska,...).
- Each genre must be played at most once per day.
- Each DJ has given you a list of genres they are willing to play.
- Each DJ can play multiple sets per day, but at most three sets during the entire event.

Suppose there are n candidate DJs and g different musical genres available. Give an efficient algorithm that either assigns a DJ and a genre to each of the 3k sets, or correctly reports that no such assignment is possible.

6 Greedy (1.5 points)

We have n skiers with heights $p_1, \ldots, p_n$ and n pairs of skis with heights $s_1, \ldots, s_n$. We want to minimize the average difference between the height of a skier and the skis that are assigned to him/her. That is, if skier p_i is assigned pair of skis $s_{a(i)}$ then we want to minimize:

$$\frac{1}{n}\sum_{i=1}^{n}(\mid p_i - s_{a(i)} \mid).$$

1. Consider the following greedy algorithm. Find the skier and the pair of skis whose height difference is minimal. Assign this skier this pair of skis. Repeat the process until every skier has skis. Prove or disprove that this algorithm is correct.

2. Consider the following greedy algorithm. Give the shortest skier the shortest pair of skis, the second shortest skier the second shortest pair of skis, the third shortest skier the third shortest pair of skis, etc. Prove or disprove that this algorithm is correct

Hint: One of the above greedy algorithms is correct and the other is not. (Of course you are not allowed to use this hint in your proof, otherwise there wouldn't be much to prove once you find a counterexample for one of the algorithms.)

7 NP completeness (1.5 points)

The problem AllOrNothing3Sat asks, given a Boolean expression that is the conjunction of clauses such that each clause contains exactly three literals, whether there is an assignment to the variables such that each clause either has three True literals or has three False literals.

1. Describe a polynomial-time algorithm to solve AllOrNothing3Sat.

2. But 3Sat is NP-hard! Why doesnt the existence of this algorithm prove that P=NP?