Technical University of Denmark

Written examination, May 17, 2022.

Course name: Algorithms and Data Structures 1

Course number: 02105

Aids: all aids, open internet.

Duration: 2.5 hours.

Weight: Exercise 1 - 35%, exercise 2 - 25%. The weight is approximate. The grade is based on an overall assessment of the multiple-choice part, the implementation part, and the written part of the exam.

Instructions for exercise 1.

Write your answers to the exercises and submit them using the digital exam system. Your answer should be in the form of a `pdf`-file containing exactly 1 page in the a4-format. The top of the page should clearly list your full name and your study id. The rest of page should contain your answers to the each of the exercises. Number each of your answers with the number of the corresponding exercise. Use a clearly readable font of size 10pt or more and margins of 2cm or more.

Asymptotic bounds should be as tight as possible. Unless otherwise specified, the base of all logarithms is 2 and $\log^k n$ means $(\log n)^k$.

"Give an algorithm" means that you should describe your solution in a short, precise, and unambiguous manner and analyze the complexity of your solution. Unless specified otherwise, the description should be in natural language and not pseudocode. The analysis should explain how you derived the complexity bound.

"Argue correctness of your algorithm" means that you should provide a short argument for correctness of your algorithm.

"Analyze the running time of your algorithm in the relevant parameters (parameters x, y, ..) of the problem" means that you should analyze the running time using the explicitly stated parameters of the problem (parameters x, y, ..).

Instructions for exercise 2.

Read the exercise text in the CodeJudge system and implement a solution. Upload your solution to the CodeJudge system.

1 Downhill Skiing

We are interested in planning trip down a mountain at a ski resort. The ski resort consists of x *positions* $p_0, \dots, p_{x-1}$ and y *ski slopes* $s_0, \dots, s_{y-1}$. Each ski slope s is defined by the following:

- a *start position* $\mathsf{start}(s)$,
- an *end position* $\mathsf{end}(s)$, and
- a *completion time* $\mathsf{time}(s)$ that indicates how many minutes it takes to traverse s on ski.

1.1 Briefly describe how to model a ski resort as a graph.

1.2 We are interested in determining if it is possible to get down the mountain before the ski slopes close. A *trip* from position a to position b is a sequence of positions starting in a and ending in b that are pairwise connected by ski slopes (i.e., each position p in the sequence is connected to the next position p' by a ski slope s such that $p = \mathsf{start}(s)$ and $p' = \mathsf{end}(s)$. The completion time of a trip is the sum of the completion time of each of the ski slopes of the sequence.

Give an algorithm that given a description of a ski resort and a time threshold t determines whether or not there is a trip from p_0 to p_{x-1} with completion time at most t. Argue correctness of your algorithm. Analyze the running time of your algorithm in terms of parameters x, y, and/or t.

1.3 We now also store a *level* for each ski slope s between, denoted $\mathsf{level}(s)$, that indicates the difficulty of traversing s. A high level for s indicates of that s is difficult to traverse. Each level is an integer between 1 and x. Give an algorithm, that given a ski resort and a time threshold t determines the smallest level ℓ such that there is a trip from p_0 to p_{x-1} with completion time at most t that only uses ski slopes of level at most ℓ. Argue correctness of your algorithm. Analyze the running time of your algorithm in terms of parameters x, y, t, and/or ℓ.

2 Implementation Exercise

Read the exercise text in the CodeJudge system.