Welcome to 02110. All practical information about the course can be found on the course webpage.

Structure

The course runs on Thursdays 8-12. Exercise class is from 8.00-10.00 and lectures from 10.00-12. There is a weekplan for each week containing information about literature for the week and exercises. The weekplans are placed under the week, where the lecture in the subject is given. Then you work with the exercises at the exercise class the week after.

The class is structured as follows:

- *8.00-10.00 Group work*. Time to work on the exercises you couldn't solve at home. The TAs will be there to help you.
- *10.15-12.00 Lecture* Including walk-through of selected exercises.

Lecture and litterature

For reading material etc for the first lecture see weekplan 1.

Exercises

The exercises on this week plan is for the exercise class in Week 1 at 8:00, i.e., before the first lecture.

1 Admin Read all the information about the course on the course webpage and on DTU Learn.

2 Union and Intersection Suppose you are given two sorted lists A and B of integers of length n_A and n_B, respectively.

2.1 Describe an algorithm that computes the set $C = A \cap B$. The output should be a sorted list C containing all elements that are in both A and B.

2.2 Describe an algorithm that computes the set $C = A \cup B$. The output should be a sorted list C containing all elements that are in either A and B.

3 Pancake sorting Suppose you are given a stack of n pancakes of different sizes. You must sort the pancakes so that smaller pancakes are on top of larger pancakes. The only operation you can perform is a flip: insert a spatula under the top k pancakes, for some integer k between 1 and n, and flip them all over.

3.1 Describe an algorithm to sort an arbitrary stack of n pancakes using $O(n)$ flips. Exactly how many flips does your algorithm perform in the worst case?

3.2 For every positive integer n, describe a stack of n pancakes where your algorithm from 2.1 requires $\Omega(n)$ flips to sort.

4 Subsequence A string P is a *subsequence* of string T if we can obtain P from T by removing 0 or more characters in T. For instance, `aba` is a subsequence of `bcadfbbba`.

4.1 Give a linear time algorithm that decides wether a string P is a subsequence of a string T.

4.2 Given k strings $P_1, P_2, \ldots, P_k$, and a string T, give an algorithm that decides which of the k strings that are subsequences of T.

5 Interval Scheduling In the interval scheduling problem you are given a set of requests labeled $1, \dots, n$. Each request has a starting time s_i and a finish time t_i. A subset of requests are *compatible* if no two of them overlap in time. The goal is to select a subset S of compatible intervals that is as large as possible. Give an algorithm that solves the interval scheduling problem.

6 Networks Consider a diagram of a telephone network which consists of switching centres and communication lines. Each communication goes between a pair of switching points, and has an associated bandwidth and a cost.

6.1 Give an algorithm that given two switching centres a and b, outputs the cost of a minimum cost path between a and b.

6.2 The bandwidth of a path of communication lines between two centres is the lowest bandwidth on the path. Give an algorithm that given two switching centres a and b outputs the maximum bandwidth of a path between a and b.

Puzzle of the week: 99 Cops A town has 99 cops. A cop is either honest or corrupt, the majority of the cops is honest. You need to figure out all the corrupt cops, with less than 299 questions. All cops know who is honest and who is corrupt, but only honest cops will answer truthfully. Corrupt cops may lie arbitrarily. For security reasons you can only ask one type of question: You may ask cop X whether cop Y is corrupt. This question will by answered by X with either "Y is corrupt" or "Y is honest".