ALGORITHM DESIGN

Exercise Sheet 1

Lecturer: Prof. Stefano Leonardi
Teaching Assistants:
Philip Lazos, Rebecca Reiffenhäuser

Academic Year 2020/2021
November 16, 2020
Due Date: December 5, 2020

General Terms: You can work on the exercises in teams of two people, and hand them in accordingly. However, make sure that each one of you has fully understood every solution you present. Do not copy any work from other students, the internet or other sources, and do not share your work with others outside your team. If at any point, any part of the exercises you hand in is apparent to be a copy of other work, this will result in the following consequences: All of your exercises, previous as well as upcoming ones, will be treated as if you did not hand them in at all, and you will have to participate in the written exam to make up for this. Please note that there will be no exception made, even if you are the original author of work someone else copied, or if your exercise partner is the one responsible. Therefore, please make sure to only choose a partner that you trust, and do not hand out your exercise solutions to others.
Late Policy: If you hand in your exercises after the due date, each day that you are late will result in a discount to your score, i.e. you will only receive 90% of the points if you are one day late, 80% on the second day after the due date, and 70% on the third. If you are late by more than three days, the assignment will get zero points in total.
Formal Requirements: Please typeset your solutions (11 pt at least), and state the names of both team members at the beginning of each sheet you hand in. Make sure to name the exact exercise each part of the solution refers to, otherwise it will not be graded. Please start a new page for each main exercise in the assignment (i.e. exercise 1, 2, etc., but not for each subquestion in them). Make sure your solution takes no more than one page for each main exercise, because everything after the first page will not be taken into account. So, for example, when the assignment has four exercises, please hand in four pages, one for each exercise. All solutions have to be sent via email to

lazos@diag.uniroma1.it **or** rebeccar@diag.uniroma1.it.

Please use the subject line "AD 2020/2021 Exercise 1" and include a *single* pdf file containing all the solutions (as well as your names, exercise set, course etc) using the filename "lastname1_lastname2_ex1.pdf".
Office Hours: There will be special hours for questions about the exercises announced via the piazza site. Presumably, this will take place in form of a zoom meeting due to the Covid19-measures.

Exercise 1

You are given a long string of letters, $w = (w_1, \dots, w_n)$. You are interested in the longest palindrome, i.e. the longest sequence $P = (w_i, \dots, w_j)$ that is the same forwards and backwards, meaning $(w_i, w_{i+1}, w_{i+2}, \dots, w_{j-1}, w_j) = (w_j, w_{j-1}, w_{j-2}, \dots, w_{i+1}, w_i)$.

1. Give an algorithm that in $O(n^2)$ time outputs the longest palindrome, and prove its running time.
2. Give an algorithm for the case that the palindrome does not have to be a continuous subsequence of w, i.e. the case that it is any sequence $P = (w_{i_1}, w_{i_2}, \dots, w_{i_j})$ for which $(w_{i_1}, w_{i_2}, \dots, w_{i_j}) = (w_{i_j}, w_{i_{j-1}}, \dots, w_{i_1})$ and $i_1 < i_2 < \dots < i_j$. Show that the algorithm solves the problem in time $O(n^3)$.

Exercise 2

You are organizing an event where investors can meet founders of young startups during an evening of interesting presentations and good food. You plan to equip the room with round tables of at least three people each, and to enable fruitful conversations, you plan on arranging the seating such that each investor $i \in I$ has only founders $f \in F$ as neighbors and vice versa. At the same time, since you know the interests of each investor, you have narrowed down the *good* pairs (i, f) into a list $P \subseteq I \times F$. Design a flow network N such that finding a maximum flow in N solves the problem of whether it is possible to find a seating arrangement with only *good* pairings. Make sure you give a formal definition of the vertex and edge set of N, and show the correctness of your construction.

Exercise 3

As a freelance programmer, you work via a website that offers projects. The website has n projects $p \in P$ that currently interest you, but in order to work on each, you need to have a certain *credit score* c_p to demonstrate the suitability of your skills. Your current score is C. However, working on each project will also influence your credit by some amount b_p (positive or negative), so after doing one project, your suitability for some of the others might have changed. You can assume that for any p, $c_p + b_p \geq 0$. Give an algorithm that finds out whether you can do all n projects in P in some order, or not. Your algorithm must run in time $O(n \log n)$. Prove your algorithm's correctness and running time.

Exercise 4

You are tasked with finding the best COVID-19 cure. So far, n candidates have been developed, with codenames $c_1, c_2, \ldots, c_n$. To test a single cure, you can add a small amount of it to a vial containing a viral solution. All of them are somewhat effective, but might require a different dose to kill the virus: for example, it could take $a_4 = 42$ units of c_4 but only $a_{31} = 6$ units of c_{31}. After putting a certain amount of a cure in a test vial, to decide if the virus has been eliminated you need to perform a complicated test which requires rare and expensive reagents. Since the world will need a very large amount of the cure, you want to find the most effective one with high precision: i.e., you want the cure $c_i \in \{c_1, \ldots, c_n\}$ such that a_i is minimal. Specifically, you already know that d units of any cure will kill the virus and want to find the best one. Describe an algorithm to solve this task, using as few tests as possible.

1. Find a deterministic algorithm that uses $O(n \log d)$.
2. Find a randomized algorithm that uses $O(n + \log n \cdot \log d)$ tests.