CSOR W4246
Summer A, 2021
Eleni Drinea

Homework 2 — Theoretical part (140 points)

Out: Friday, May 21, 2021
Due: 11:59pm, Friday, May 28, 2021

Homework Instructions.

1. For all algorithms that you are asked to "give" or "design", you should
   - Describe your algorithm clearly in English.
   - Give pseudocode.
   - Argue correctness.
   - Give the best upper bound that you can for the running time.
   - You're encouraged to analyze space but points will not be deducted unless the problem explicitly asks for a space analysis.
2. **If you give a Dynamic Programming algorithm, the above requirements are modified as follows:**
   (a) Clearly define the subproblems in English.
   (b) Explain the recurrence in English. (This counts as a proof of correctness; feel free to give an inductive proof of correctness too for practice but points will not be deducted if you don't). Then give the recurrence in symbols.
   (c) State boundary conditions.
   (d) Analyze time.
   (e) Analyze space.
   (f) If you're filling in a matrix, explain the order to fill in subproblems in English.
   (g) Give pseudocode.
3. **You should not use any external resources for this homework.** Failure to follow this instruction will have a negative impact on your performance in the exam (and possibly in interviews). For the same reason, **you should avoid collaborating** with your classmates, at least when working on problems 1, 3 and 4. I also encourage you to work on every recommended exercise.
4. You should submit your assignment as a **pdf** file to Gradescope. Other file formats will not be graded, and will automatically receive a score of 0.
5. I recommend you type your solutions using LaTeX. For every assignment, you will earn 5 extra credit points if you type your solutions using LaTeX or other software that prints equations and algorithms neatly. If you do not type your solutions, make sure that your hand-writing is very clear and that your scan is high quality.
6. You should write up the solutions **entirely on your own**. Collaboration is limited to discussion of ideas only. You should adhere to the department's academic honesty policy (see the course syllabus). Similarity between your solutions and solutions of your classmates or solutions posted online will result in receiving a 0 in this assignment, and possibly further disciplinary actions. There will be no exception to this policy and it may be applied retro-actively if we have reasons to re-evaluate this homework.

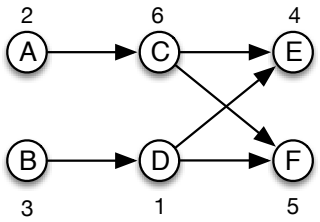
Homework Problems

1. (15 points) Given an undirected graph $G = (V, E)$ and a specific edge $e \in E$, give an efficient algorithm that determines whether G has a cycle that contains e.

2. (35 points) You are given a directed graph $G = (V, E)$ in which each node $u \in V$ has an associated price p_u which is a positive integer. Define the array `cost` as follows: for each $u \in V$,

   $$\texttt{cost}[u] = \text{price of the cheapest node reachable from } u \text{ (including } u \text{ itself)}$$

   For instance, in the graph below (with prices shown for each vertex), the `cost` values of the nodes A, B, C, D, E, F are $2, 1, 4, 1, 4, 5$ respectively.

   Your goal is to design an algorithm that fills in the entire array `cost`.

   (a) (20 points) Give a linear-time algorithm that works for directed *acyclic* graphs.

   (b) (15 points) Extend this to a linear-time algorithm that works for all directed graphs.

3. (25 points) You want to determine the probability of obtaining exactly k heads when n biased coins are tossed independently, where the i-th coin has known probability p_i of coming up heads.

   Given n, k, and $\{p_1, \ldots, p_n\}$, design and analyze an algorithm for this task that runs in time $O(nk)$. (Assume that you can multiply and add two numbers in $[0, 1]$ in $O(1)$ time.)

4. (30 points) A hotel chain is considering opening a series of hotels along a long highway. The n possible locations are on a straight line, and the distances of these locations from the start of the highway are $m_1, m_2, \ldots, m_n$, in increasing order of miles. The constraints are

   - At each location, at most one hotel may open. The expected profit from opening a hotel at location i is $p_i > 0$.
   - Any two hotels should be at least k miles apart, for some integer $k > 0$.

   Give an efficient algorithm to compute the maximum expected total profit subject to the given constraints. *Hint: Try to give a $o(n^2)$ algorithm.*

5. (35 points) There is a number pyramid. The top floor has 1 positive integer. The second floor has 2 positive integers, the third floor 3, and so on and so forth.

   Here is an example of a pyramid with 4 floors:

   6
   3 8
   7 1 0
   2 6 4 4

   You start from the top floor. At each step, you can go either diagonally down to the left or diagonally down to the right. This way you can form a *route* from the top of the pyramid to its base. Note that there are multiple routes that end somewhere in the base of the pyramid. Your goal is to maximize the sum of the integers on any route.

   Design and analyze an efficient algorithm to compute the maximum sum on any *route* from the top to the base of the pyramid. You should also return an optimal route. You may assume that there are n floors in the pyramid.

   In the above example, the maximum sum is 22 achieved by route $6 \to 3 \to 7 \to 6$.

RECOMMENDED exercises: *do NOT return, they will not be graded.*

1. A server has n customers waiting to be served. The service time for customer i is t_i minutes. So if the customers are served in order of increasing i, the i-th customer spends $\sum_{j=1}^{i} t_j$ minutes waiting to be served.

   Given n, $\{t_1, t_2, \ldots, t_n\}$, design an efficient algorithm to compute the optimal order in which to process the customers so that the total waiting time below is minimized:

   $$T = \sum_{i=1}^{n} \text{(time spent waiting by customer } i\text{)}$$

2. `https://leetcode.com/problems/climbing-stairs/`. Contributed by Michael Jan; everyone is encouraged to work on this problem before attempting the homework problems.

3. Give an efficient algorithm to compute the length of the **longest** increasing subsequence of a sequence of numbers $a_1, \ldots, a_n$. A subsequence is any subset of these numbers taken in order, of the form $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ where $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and an increasing subsequence is one in which the numbers are getting strictly larger.

   For example, the longest increasing subsequence of $5, 2, 8, 6, 3, 6, 7$ is $2, 3, 6, 7$ and its length is 4.

4. Consider an array A with n numbers, some of which (but not all) may be negative. We wish to find indices i and j such that

$$\sum_{k=i}^{j} A[k]$$

is maximized. Give an efficient algorithm for this problem.

5. Given two strings $x = x_1 x_2 \cdots x_m$ and $y = y_1 y_2 \cdots y_n$, we wish to find the length of their longest common substring, that is, the largest k for which there are indices i and j such that

$$x_i x_{i+1} \cdots x_{i+k-1} = y_j y_{j+1} \cdots y_{j+k-1}.$$

Give an efficient algorithm for this problem.