

CS6033 – Design and Analysis of Algorithms I

Homework 5

Manuel — NYU (Spring 2021)

Reminders

- Write in a neat and legible handwriting or use \LaTeX
- Clearly explain the reasoning process
- Write in a complete style (subject, verb, and object)
- Be critical on your results

*Questions preceded by a * are optional. Although they can be skipped without any deduction, it is important to know and understand the results they contain.*

Ex. 1 — The partition problem

- * 1. Explain what the *linear partition problem* is and where it arises.
- 2. A simple strategy consists in computing the average size of a partition and then insert a divider close to this average. Is this a good solution? Explain, or provide a counter example.

Let n be the number of elements and k be the number of ranges. We denote by $M(n, k)$ the minimum cost over all the partitioning of the n elements into k ranges, where the cost is the largest sum of elements in one of its parts.

- 3. Provide a recursive algorithm to compute $M(n, k)$. Its detailed pseudocode is not required here.
- 4. What is the complexity of the previous algorithm?
- 5. What quantities could be stored and not recomputed over and over again?
- 6. Provide the pseudocode of a dynamic programming algorithm solving the partition problem.
- 7. Prove the correctness of this new algorithm.
- 8. Show that the complexity is now $\mathcal{O}(kn^2)$.
- 9. As knowing the minimal cost is often useless in practice, we desire to reconstruct the path such as to know where to set the dividers. Explain how this can be achieved by a simple subroutine.

Ex. 2 — Critical thinking

Given a black box B which returns a random integer in the range $0 - 4$, create a random number generator whose output is a random integer in the range $0 - 7$. Extend this idea and write an algorithm which given an integer n , uses B to generate random integers in the range $0 - n$. Is there any restriction on n ? All the random number generators are expected to follow the uniform distribution.

Ex. 3 — Bellman-Ford algorithm

In the lectures, Bellman-Ford algorithm (3.17|3.157) expects a weighted graph with no negative cycle. Write an algorithm to determine whether or not a given graph features a negative cycle.

* Ex. 4 — Augmenting path

Complete the proof of lemma (4.13|4.183).

Ex. 5 — *Wifi network*

On campus there are k internet hotspots. Their positions are known and defined as simple planar coordinates. They also feature a range parameter r within which a user can connect, and a load l defining the maximum number of users the base can serve. Design a polynomial time algorithm which given r , l , the location of the k hotspots, and the position of n clients decides whether they can all connect to internet. Provide the pseudocode for the algorithm, proof its correctness, and complexity.

* **Ex. 6 —** *Interview problem*

Let $M = (m_{i,j})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ be an $m \times n$ matrix of positive integers such that the $m_{i,j}$, $1 \leq i \leq m$ and $1 \leq j \leq n$ are in ascending order

- from left to right in each row;
- from top to bottom in each column;

Given an integer p , return its (i, j) coordinates if it is in M and -1 otherwise.

Example. Given $M = \begin{pmatrix} 1 & 4 & 7 & 11 & 15 \\ 2 & 5 & 8 & 12 & 19 \\ 3 & 6 & 9 & 16 & 22 \\ 10 & 13 & 14 & 17 & 24 \\ 18 & 21 & 23 & 26 & 30 \end{pmatrix}$, for 5 is at $(2, 2)$ and for 20 return -1 .