Theorey Assignment-1: ADA Winter-2023

Full Marks: 60
Deadline: February 23 at 11:59 pm.

General instructions: While solving the assignment, you are encouraged to discuss with your classmate and/or senior if you want to. In case you do so, please mention the name (and also roll number in case of classmate) of the persons you have discussed with. If you look-up the solution in the internet, you must provide the link at which you have seen the solution. Finally, write the solutions in your own language. Copying solution will be reported as plagiarism. You are allowed to use anything that you have seen in the class or tutorial or in some earlier course including DSA and/or Discrete Mathematics.

Other Instructions: For each question, if your solution uses dynamic programming, we need you to provide the following:

(a) If there is any preprocessing stage before mentioning subproblem definition, state that. If the preprocessing algorithm uses something known from some tutorial or some earlier course, e.g. DSA, then just mention that as a subroutine. No need to describe that.

(b) A precise description of the subproblems you want to solve in the dynamic program. *Notice this is just the subproblem, not how to solve it, or how it was obtained.*

(c) A recurrence which relates a subproblem to "smaller" (whatever you define "smaller" to mean) subproblems. *Notice this is just the recurrence, not the algorithm or why the recurrence is correct.*

(d) Identify the subproblem that solves the original problem

(e) A description of your dynamic programming algorithm which solves the recurrence efficiently. The description must be in English containing mathematical symbols or it can be a pseudocode in plain text. A C++/Java program etc with variable declartion etc will not be accepted. *You do not need to prove correctness of the pseudocode*

(f) An argument for the running time of your dynamic algorithm.

For each question, if your solution uses divide and conquer paradigm, then we need you to provide the following.

(a) If there is any preprocessing stage before mentioning subproblem definition, state that. If the preprocessing algorithm uses something known from some tutorial or some earlier course, e.g. DSA, then just mention that as a subroutine. No need to describe that.

(b) A precise description of the subproblems, i.e. how you divide the actual problem into smaller subproblems.

(c) A precise description on *how you combine the solutions of the subproblems* to solve the actual problem.

(d) A description of your complete recursive algorithm. You can refer to the subroutines of (a) and (b). The description must be in English containing mathematical symbols or it can be a pseudocode in plain text. A C++/Java program etc with variable declartion etc will not be accepted.

(e) Analysis of running time of the algorithm including a recurrence relation and the solution the recurrence relation.

Question 1 *Consider the problem of putting L-shaped tiles (L-shaped consisting of three squares) in an $n \times n$ square-board. You can assume that n is a power of 2. Suppose that one square of this board is defective and tiles cannot be put in that square. Also, two L-shaped tiles cannot intersect each other. Describe an algorithm that computes a proper tiling of the board. Justify the running time of your algorithm. (***15 Marks***)*

Question 2 *Suppose we are given a set L of n line segments in 2D plane. Each line segment has one endpoint on the line $y = 0$, one endpoint on the line $y = 1$ and all the $2n$ points are distinct. Give an algorithm that uses dynamic programming and computes a largest subset of L of which every pair of segments intersects each other. You must also give a justification why your algorithm works correctly. (***20 Marks***)*

Question 3 *Suppose that an equipment manufacturing company manufactures s_i units in the i-th week. Each week's production has to be shipped by the end of that week. Every week, one of the three shipping agents A, B and C are involved in shipping that week's production and they charge in the following:*

- *Company A charges a rupees per unit.*
- *Company B charges b rupees per week (irrespective of the number of units), but will only ship for a block of 3 consecutive weeks.*
- *Company C charges c rupees per unit but returns a reward of d rupees per week, but will not ship for a block of more than 2 consecutive weeks. It means that if s_i unit is shipped in the i-th week through company C, then the cost for i-th week will be $cs_i - d$.*

*The total cost of the schedule is the total cost to be paid to the agents. If s_i unit is produced in the i-th week, then s_i unit has to be shipped in the i-th week. Then, give an efficient algorithm that computes a schedule of minimum cost. (Hint: use dynamic programming) (***25 Marks***)*