Theorey Assignment-2: ADA Winter-2023

Full Marks: 50, **Deadline:** April 8 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 or institute senior) 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 free to use some algorithm as subrouting taught in the class or in tutorial.

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.

(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 with instructions in plain text. A C++/Java program etc with variable declaration 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.

If your solution uses application of graph traversal, then your answer must have the following structure.

(a) First describe how you formulate it as a graph theoretic problem (if applicable). This needs to clearly specify how the construction of graph works. If possible give one example and explain how construction works.

(b) Then, explain how you use graph traversal algorithms, e.g. DFS/BFS of directed/undirected graphs and any other properties taught in the tutorial to solve this problem. Give a precise description of the algorithm. For example, if you want to use how to find all the cut vertices of an undirected graph in linear time, then you can mention that without describing it.

(c) Write down the necessary and sufficient conditions that explains why your algorithm works correctly.

(d) Give a proper explanation of the running time of your algorithm.

Question 1 *The police department in the city of Computopia has made all the streets one-way. But the mayor of the city still claims that it is possible to legally drive from one intersection to any other intersection.*

*(a) Formulate this problem as a graph theoretic problem and explain why it can be solved in linear time. (**7 Marks**)*

*(b) Suppose it was found that the mayor's claim was wrong. She has now made a weaker claim: "if you start driving from town-hall, navigating one-way streets, then no matter where you reach, there is always a way to drive legally back to the town-hall. Formulate this weaker property as a graph theoretic problem and explain how it can be solved in linear time. (**8 Marks**)*

Question 2 *Given an edge-weighted connected undirected graph $G = (V, E)$ with $n + 20$ edges. Design an algorithm that runs in $O(n)$-time and outputs an edge with smallest weight contained in a cycle of G. You must give a justification why your algorithm works correctly. (**15 Marks**) (**Hint:** Formulation into graph theoretic problem is not necessary as the graph is given already.)*

Question 3 *Suppose that G be a directed acyclic graph with following features.*

- *G has a single source s and several sinks $t_1, \ldots, t_k$.*
- *Each edge $(v \to w)$ (i.e. an edge directed from v to w) has an associated weight $p(v \to w)$ between 0 and 1.*
- *For each non-sink vertex v, the total weight of all the edges leaving v is* $\sum_{(v\to w)\in E} Pr(v \to w) = 1$.

*The weights $Pr(v \to w)$ define a random walk in G from the source s to some sink t_i; after reaching any non-sink vertex v, the walk follows the edge $v \to w$ with probability $Pr(v \to w)$. All the probabilities are mutually independent. Describe and analyze an algorithm to compute the probability that this random walk reaches sink t_i for every $i \in \{1, \ldots, k\}$. You can assume that an arithmetic operation takes $O(1)$-time. (see Figure 1 for an illustration) (**20 Marks**)*

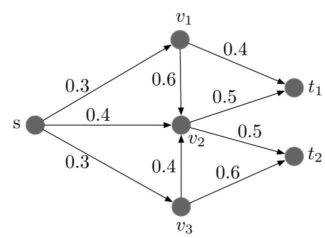

Figure 1: An illustration of Question 3. The probabilities for each of the directed edges are given. The probability that a random walk uses the path $s \to v_1 \to t$, then the probability that the walk reaches from s to t using this path is $0.3 \times 0.4 = 0.12$. Similarly, for the path $s \to v_1 \to v_2 \to t$, the probability that the random walk reaches from s to t using this path is $0.3 \times 0.6 \times 0.5 = 0.09$.