TOBB ETÜ, BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL 331/531 Design and Analysis of Algorithms

HOMEWORK 1 (100 Points)

Due Date: March 26, 2024

The specifics of implementation and submission will be announced in piazza soon.

1 [20 POINTS] COMPUTING THE NUMBER OF SHORTEST PATHS

You will write a Java program that takes an unweighted undirected graph $G = (V, E)$ represented as a collection of adjacency lists, and a designated vertex $s \in V$ as input. Your program should compute the number of distinct shortest paths to each vertex. Your program should return an array A of size $|V|$ such that $A[i]$ contains the number of distinct shortest paths to vertex i. If the running time of your algorithm is not $O(n+m)$, it will not pass the time limit. Note: We say that tho paths are distinct if they have at least one uncommon edge.

2 [20 POINTS] FINDING ARTICULATION POINTS

Given an conected undirected graph $G = (V, E)$, a vertex $u \in V$ is called an *articulation point*, if the removal of v from G makes G disconnected. You are asked to write a Java program that takes an unweighted undirected and connected graph $G = (V, E)$ as input and finds all the articulation points it has. Your program should return an integer array that contains the articulation points. If the running time of your algorithm is not $O(n+m)$, it will not pass the time limit. **Your program shall not be recursive.**

3 [20 POINTS] FINDING THE GLOBAL SINK

Given a directed graph $G = (V, E)$, we say that $v \in V$ is a *global sink*, if $(u, v) \in E$ for any $u \in V \setminus v$, and $(v, u) \notin V$ for any $u \in V$. That is, v does not have any outgoing edges, and it has an

incoming edge from every other vertex.

You are asked to write a Java program that takes a directed graph $G = (V, E)$ represented as an adjacency matrix as input. Your program should return the index of the global sink if one exists, and it should return -1 otherwise. If the running time of your algorithm is not $O(n)$, it will not pass the time limit.

4 [20 POINTS] FINDING THE LENGTH OF THE SHORTEST PATH

You will write a Java program that takes the following input: a weighted directed graph $G = (V, E)$ represented as an adjacency list, and two designated vertices s and t. Your program should return the length of the shortest path from s to t. You can assume that all edge lengths are nonnegative. If there is no path from s to t, then your algorithm should return double.MAX_VALUE. If the running time of your algorithm is not $O(\min\{n^2, m \cdot \lg n\})$, it will not pass the time limit.

5 [20 POINTS] WHO IS TO BE MATCHED WITH WHOM?

Recall that at some point Utku had invented a time machine and went to Roman times. You were at the beginning of your college life and most probably focused on decrypting texts encrypted with Ceaser cipher. Almost nobody shown interest in the details of the construction of the time machine. Utku has also terminated his studies on the time machine since he was quite scared of being trapped in a different time.
Mr. Bidik, however, was quite interested in the specifics of the time machine and since then have been heavily working on constructing a more stable version. He has finally managed to make the machine work properly. Recently, he traveled to year 4100AC and came back. He has given a lot of information to us about the future.
He had the chance to go to some of the lectures at a university. He said that Artificial Intelligence Engineering Departments are quite popular in the future. In the finance classes he attended, people were discussing the bankruptcy of Meta due to mis-speculation on the future of metaverse. He has also attended the algorithms classes while he was there. To his surprise, people were still interested in algorithms that match men and women. This was partly because a deadly world war, which put the future of humanity at risk.
The most fundamental issue -in the future- was to increase the size of the population and thus, when matching men and women, they did not care about the individual preferences. Rather, the problem they studied was as follows:

As input, we are given an integer array M of size n representing the ages of men in the society, and an integer array W of size n representing the ages of women in the society, and an integer k. The goal is to obtain a maximum sized matching between men and women such that no person is matched with someone whose age differs by more than k. For instance, if $k = 36$, somebody at the age of 1286 can be matched with someone with age $1250 - 1322$.
The output of your program should be a single integer: the size of the maximum matching.

PS: Due to advancement in the life sciences, people can live up to $2^{31} - 1$ years.