CS331: Algorithms and Complexity
Homework III

Trung Dang Ryan Park Kevin Tian

Due date: October 16, 2024, end of day (11:59 PM), uploaded to Canvas.

Late policy: 15% off if submitted late, and 15% off for every further 24 hours before submission.

Please list all collaborators on the first page of your solutions.

When runtimes are unspecified, slower runtimes than the intended solution receive partial credit.

1 Problem 1

We say that directed graph $G = (V, E)$ is *moderately connected* if for all pairs of distinct vertices $(u, v) \in V \times V$, either v is reachable from u, or u is reachable from v (not necessarily both). Give an algorithm that determines whether an input directed graph G is moderately connected.

2 Problem 2

Let $G = (V, E)$ be a connected undirected graph with $n := |V|$ and $m := |E|$, and let $s \in V$ be a source vertex. For each $v \in V$, define $c(v)$ to be the number of distinct shortest paths between s and v. Give an $O(m + n)$-time algorithm that takes as input G and computes $c(v)$ for all $v \in V$.

3 Problem 3

Define the *width* of a path P to be the minimum edge weight along the path.[1] Let $G = (V, E, \mathbf{w})$ be a directed graph with $n := |V|$ and $m := |E|$, and let $s \in V$ be a source vertex. Assume all of V is reachable from s. For each $v \in V$, define $w(v)$ to be the maximum width of a path from s to v. Give an $O(m \log(n))$-time algorithm that takes as input G and computes $w(v)$ for all $v \in V \setminus \{s\}$.

Here, we consider the width of an empty path from s to itself to be ∞.

Your algorithm should be a very small modification of Dijkstra's algorithm (Section 3.1, Part V).

4 Problem 4

Alison the Square Hole Girl has n objects (numbered from 1 to n) and m boxes (numbered from 1 to m). For each object $i \in [n]$, Alison is given a set $S[i] \subseteq [m]$, where object i fits into the opening hole of box $j \in [m]$ iff $j \in S[i]$. We call j compatible with i in this case. Further, for all boxes $j \in [m]$, Alison knows that it can only contain at most $C[j]$ different objects, for some integer $C[j]$.

Give an algorithm that takes as input all of the $\{S[i]\}_{i \in [n]}$ (given as an Array of Array instances), and C (given as an Array instance), and either outputs a way for Alison to put all objects into compatible boxes, or concludes no such compatible assignment of objects to boxes exists.

For this problem, proving a runtime that is polynomial in m and n receives full credit.

[1] Intuitively, you can think of each edge weight as specifying the diameter of a tube, so that the minimum edge weight along the path capacitates the width of the tubes, i.e., the largest amount of material that can pass.

5 Problem 5

Complete the assignment at this link. This link is only accessible on your UT email.