# CS 170 Homework 4

Due **2021-02-16, at 10:00 pm**

## 1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, write "none".

## 2 True Source

Design an efficient algorithm that given a directed graph $G$ determines whether there is a vertex $v$ from which every other vertex can be reached. (Hint: first solve this for directed acyclic graphs. Note that running DFS from every single vertex is not efficient.)

**Please give a 3-part solution to this problem.**

## 3 Finding Clusters

We are given a directed graph $G = (V, E)$, where $V = \{1, \dots, n\}$, i.e. the vertices are integers in the range 1 to $n$. For every vertex $i$ we would like to compute the value $m(i)$ defined as follows: $m(i)$ is the smallest $j$ such that vertex $j$ is reachable from vertex $i$. (As a convention, we assume that $i$ is reachable from $i$.) Show that the values $m(1)$, …, $m(n)$ can be computed in $O(|V| + |E|)$ time.

**Please give a 3-part solution to this problem.**

## 4 All Roads Lead to Rome

You are the chief trade minister under Emperor Caesar Augustus with the job of directing trade in the ancient world. The Emperor has proclaimed that *all roads lead to (and from) Rome*; that is, all trade must go through Rome. In particular, you are given a strongly connected directed graph $G = (V, E)$ with positive edge weights, and there is a particular node $v_0 \in V$ (Rome).

(a) Give an efficient algorithm that finding shortest paths between *all pairs of nodes*, with the one restriction that these paths must all pass through $v_0$ (Rome). Make your algorithm as efficient as you can (perhaps as fast as Dijkstra's algorithm). Note this algorithm only needs to return a data structure that can perform these queries; something similar to dijkstras which returns a representation of a shortest paths tree might be useful to keep in mind.

**Please give a 3-part solution.**

(b) Occasionally, Augustus will ask you for the (smallest) distance between two vertices. You want to do this as quickly as possible, so that Augustus does not have your head.

This is called a *distance query*: Given a pair of vertices $(u, v)$, give the distance of the shortest path from $u$ to $v$ that passes through $v_0$. Describe how you might store the results such that you require $O(|V|)$ storage, and you can compute the result in $O(1)$ time. For your answer, a clear description of the data structure and its usage is sufficient.

(c) On the other hand, the traders need to know the paths themselves.

This is called a *path query*: Given a pair of vertices $(u, v)$, give the shortest path from $u$ to $v$ that passes through $v_0$. Describe how you might store the results such that you require $O(|V|)$ storage, and you can compute the result in $O(|V|)$ time. Again, a clear description of the data structure and its usage is sufficient.

## 5 The Greatest Roads in America

Arguably, one of the best things to do in America is to take a great American road trip. And in America there are some amazing roads to drive on (think Pacific Crest Highway, Route 66 etc). An intrepid traveler has chosen to set course across America in search of some amazing driving. What is the length of the shortest path that hits at least $k$ of these amazing roads?

Assume that the roads in America can be expressed as a directed weighted graph $G = (V, E, d)$, and that our traveler wishes to drive across at least $k$ roads from the subset $R \subseteq E$ of "amazing" roads. Furthermore, assume that the traveler starts and ends at her home $h \in V$. You may also assume that the traveler is fine with repeating roads from $R$, i.e. the $k$ roads chosen from $R$ need not be unique.

Provide a 3-part solution with runtime in terms of $n = |V|$, $m = |E|$, $k$, and $r = |R|$.

Hint: First consider $k = 1$. How can $G$ be modified so that we can use a "common" algorithm to solve the problem?

## 6 Vertex Cut

Let $G = (V, E)$ be an undirected, unweighted graph with $n = |V|$ vertices. The *distance* between two vertices $u, v \in G$ is the length of the shortest path between them. A *vertex cut* of $G$ is a subset $S \subseteq V$ such that removing the vertices in $S$ (as well as incident edges) disconnects $G$.

Show that if there exist $u, v \in G$ of distance $d > 1$ from each other, that there exists a vertex cut of size at most $\frac{n-2}{d-1}$. Assume $G$ is connected.