CSCI-570: Homework # 3

Due on Friday, September 19 , 2014

Saket Choudhary
skchoudh@usc.edu
2170058637

Contents

HW3

(2)

A) The intersections can be viewed as the nodes of a directed graph. An intersection I_i can be reached from I_j given that there is an edge incident from I_j to I_i that is $(I_j, I_i) \in E$ where E= set of edges of Graph $G(V, E)$
If such a directed graph allows to reach from any point to any other point, it needs to be strongly connected implying there is a path from I_i to I_j and from I_j to I_i. Checking if a path from I_j exists to I_i will involve reversing the edge directions in the directed graph and checking if I_i can be reached from I_j. So if the mayor is right, it should be possible to traverse from I_j to I_i with the edges inverted.
Such a strongly connected directed graph can be traversed in linear time using DFS with a run time of O(n+m)
B) Since the mayor's original claim is false $\implies$ G is not strongly connected. However the focus shifts to the town hall being "strongly connected" with the rest of the nodes. In order for this to happen the node for town hall say T must be a sink of a strongly connected component. The approach would involve determining all such components containing T such that they are strongly connected. Next run a DFS in $O(n+m)$ to determine all nodes reachable from T if all these nodes belong to to same strongly connected component, it should be possible to travel from T to these and back.

(3: Ch#3 Ex#3)

For outputting a cycle G(if any) in G, we perform a BFS and keep track of nodes visited in an array. Initially $visitied[i] = 0 \forall i \in V$ and change the status of this array as we traverse the nodes checking $if\ visited[i] = 1;\ then\ cycle\ exists$ else we follow the following algorithm for creating a topological ordering.
BSF traversal takes O(n+m) and so does topological ordering

(4)

Distance between neighboring gas stations is p miles. Let's stop at stations $\{s_1, s_2, ...s_k\}$. To minimise the number of stops, we need to stop only if the distance to the next station is larger than what can be covered with the petrol in the car at present. As long as we can reach the $(i+1)^{th}$ petrol station, we should not stop at i^{th}.
Consider $\{s_1', s_2', s_3', ...s_k'\}$ to be the optimal solution

(5: Ch#4 Ex#3)

Cos

(6)

Given two sets A and B, each containing n positive integers. Perform reordering maximising $\Pi_{i=1}^{n} a_i^{b_i}$.
There are 4 ways to proceed.
1. Larger values of a raised to larger values of b
2. Smaller values of a raised to larger values of b
3. Larger values of a raised to smaller values of b
4. Smaller values of a raised to smaller values of b

It is easy to rule out Case 4, since it would the smallest possible product. The case maximising the payoff is Case 1 since the product is maximised when the individual terms are masimised which is possible if the larges number has the largest exponent.
It

(7: Ch#4 Ex#4)

Given two sequence S' of size m and S of size n, to determine if $S' \subset S$
S' can be visualised as a DAG or more specifically as a topological ordering. Also treat S as a DAG. We keep two pointers, one for S' and one for S and perform a DFS on S till we hit a element from S' starting with $S'[0]$, once $S'[0]$ is found in S we start a DFS again after deleting all preceding elements of S now continuing till we find $S[1]$ and deleting the intermediate hits if any.

(8)