HW 9

Due on **Wednesday, November 29, 2017 at 10am**

CS/ECE 374: Algorithms & Models of Computation, Fall 2017 Version: **1.0**

Submission instructions as in previous homeworks.

Do not use hashing or dictionary data-structures in the solutions of the questions here. Any dynamic programming solution should be done using an iterative algorithm.

1 (100 PTS.) Flood it.

(This question was inspired by the game *Open Flood* [available as an app on android].)

You are given a directed graph G with n vertices and m edges (here $m \geq n$). Every edge $\mathsf{e} \in \mathsf{E}(\mathsf{G})$ has a color $c(\mathsf{e})$ associated with it[1]. The colors are taken from a set $C = \{1, \ldots, \xi\}$ (assume $\xi \leq n$), and every color $c \in C$, has price $p(c) > 0$.

Given a start vertex $s_0 = s$, and a sequence of $\Pi = \langle c_1, \ldots, c_\ell \rangle$ of colors, a ***compliant walk***, at the ith time, either stays where it is (i.e., $s_i = s_{i-1}$), or alternatively travels a sequence of edges of color c_i that starts at s_i. Formally, if there is a path $\sigma \equiv (u_1, u_2), (u_2, u_3), \ldots, (u_{\tau-1}, u_\tau) \in \mathsf{E}(\mathsf{G})$, such that $c\big((u_j, u_{j+1})\big) = c_i$, for all j, and $u_1 = s_i$, then one can set $s_{i+1} = u_\tau$. The ***price*** of Π is $p(\Pi) = \sum_{i=1}^{\ell} p(c_i)$.

Describe an algorithm, as fast as possible, that computes the cheapest sequence of colors for which there is a compliant walk in G from a vertex s to a vertex t.

For full credit, your algorithm should run in $O(m \log m)$ time (be suspicious if you get faster running time). Correct solutions providing polynomial running time would get 50% of the points.

2 (100 PTS.) Flood it II.

We use the same framework as the previous question. Describe an algorithm, as fast as possible, that given a vertex s, and vertices $t_1, \ldots, t_k$, computes the cheapest sequence Π of colors for which there are k walks $W_1, \ldots, W_k$, such that W_i is compliant with Π, and it walks from s to t_i, for $i = 1, \ldots, k$.

Do not use dynamic programming here [it doesn't work]. Instead, think about the state of the system after i colors of the sequence were used – where would the k walks in the graph be at this point in time? Build the appropriate state graph, and solve the appropriate problem in this graph. The running time of your algorithm should be polynomial if k is a constant. As usual, you might want to start thinking about the case $k = 2$ first.

You should be able to solve this question fully even if your solution to the first question is sub-optimal.

3 (100 PTS.) Halloween!

It is Halloween, and Zaphod Beeblebrox is ready. He has a map (i.e., directed graph) of Shampoo-Banana with the houses marked, with each house v marked with how much candy $c(v) \geq 0$ one gets if visiting this house. The map also connects houses u, v by a directed edge (u, v), if Zaphod is willing to go directly from u to v. For reasons that are not well understood (maybe, Zaphod's brain-care specialist, Gag Halfrunt knows why), the fact that Zaphod is willing to go from u to v, does not imply that he is willing to go from v to u (i.e., the graph is truly a directed graph).

Given the directed graph G, describe an algorithm as fast possible (and prove its correctness), that computes the walk that starts from a vertex s, and collects as much candy as possible. The algorithm should only output the amount of candy the optimal walk collects.

[1] This is a simple directed graph – no self loops or parallel edges. Every edge has only a single color associated with it. (Unless explicitly stated otherwise, you can always assume a given directed graph is simple.)

Note, that the walk is allowed to visit the same vertex several times, but you get candy only the first time you visit it.

Also, show how to modify the algorithm so that it outputs the optimal walk. What is the length of this walk in the worst case?[2]

[Hint: What if the graph is strongly connected?]

[2] There was extensive discussion on my neighborhood mailing list after Halloween with various statistics [how many kids visited, what candies were consumed, etc] – people seems to be taking this stuff seriously.