HW 9

Due on **Wednesday, November 4, 2020 at 10am CST**

CS/ECE 374: Algorithms & Models of Computation, Fall 2020 Version: **1.11**

Submission instructions as in previous homeworks.

25 (100 PTS.) GraphCarts

GraphCarts is a new open-world virtual reality racing game. A race course in GraphCarts is modeled as a directed graph G, where each road segment is associated with a distance $w(u, v) > 0$ that describes the amount time it takes to get from intersection u to intersection v. Since the graph is directed, some roads are one-way. Even among two-way roads, it may be slower to go in one direction than another, i.e. it is possible that $w(u, v) \neq w(v, u)$. You start at a designated intersection s, and the game ends as soon as you reach the destination intersection t. By "open-world," what the developers of GraphCarts mean is that you are welcome to travel through the same intersection or along the same road segment multiple times.

The main challenge in GraphCarts is that some of the road segments contain *checkpoints* ($\varphi(u, v) = 1$ if there is a road segment with a checkpoint on it from intersection u to intersection v, $\varphi(u, v) = 0$ otherwise). The rule is that you're required to pass through at least two checkpoints (it can be the same checkpoint multiple times) *before* you reach the destination (reaching the destination without having passed through two checkpoints is an instant loss).

25.A. (25 PTS.) Give an algorithm, as fast as possible, to find the fastest path from s to t in GraphCarts, given as input a directed graph describing the race course. Briefly explain why your algorithm is correct, and analyze its running time.

25.B. (25 PTS.) In the downloadable content for GraphCarts, the player gets one free "Time Freeze" powerup that lets them traverse a single road segment in 0 time. This power-up can only be used once, so for example if they use it to travel along the road segment $u \to v$ in 0 time, and then they travel $u \to v$ again, the second takes the usual amount of time $w(u, v) > 0$. The rule about needing to pass through a checkpoint twice still applies. Modify your algorithm to find the fastest path from s to t under this new rule. Briefly explain why your algorithm is correct, and analyze its running time.

25.C. (50 PTS.) In the second downloadable content for GraphCarts, the player gets k free "Negative Time" powerup (you can assume $k < n$). A single such powerup lets them traverse a road segment in -5 time (they are allowed to use several powerups on the same edge – each such time is counted separately). They are allowed to use up to k powerups during the game.

Fortunately, the developers removed the checkpoint rule that everybody hated so much from this version of the game. Just get from s to t as fast as you can, no need to pass through any checkpoints.

Design an algorithm, as fast as possible, to compute the fastest path from s to t under this new rule. Briefly explain why your algorithm is correct, and analyze its running time.

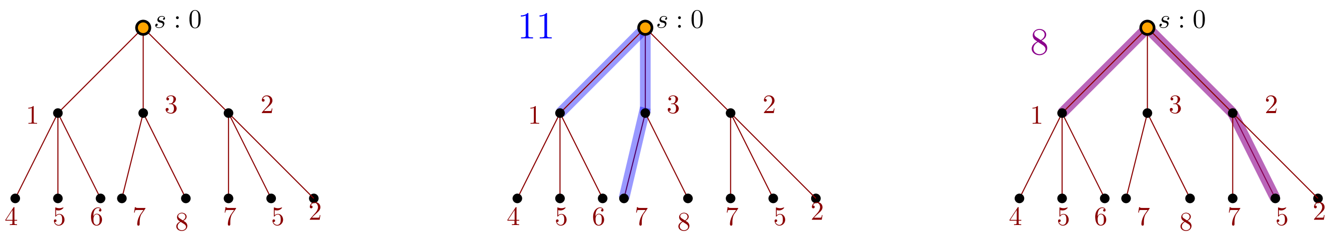
26 (100 PTS.) DynCraft

You're in charge of setting up some gold mining infrastructure within an underground cave system. The cave system can be modeled as a directed tree. It consists of N chambers. The root is the entrance to the cave system, the chamber s. Each chamber is connected to at most 3 subtrees (a ternary rooted tree!).

The quantity of gold in each chamber u is denoted by a function $g(u)$.

To mine for gold and bring it up to the entrance, you need to place conveyor belts in the corridors connecting the cave chambers. You have a limited number of L conveyor belts, $L = O(N)$. If you place a conveyor belt in a corridor connecting u to v, then it can carry gold from v to u towards the entrance. You can mine all the gold from all of the chambers that can reach the entrance through a path of conveyor belts.

The following is an example with three belts, and two different solutions. Here, the root has value 0 associated with it. Here, the first solution has a better value than the second solution.



Give a dynamic programming algorithm that determines the optimal way to place the L conveyor belts in order to mine the largest amount of gold.

27 (100 PTS.) Max Norm Suppose you have a directed graph with n vertices and m edges, where each edge e has weight $w(e)$, and no two edges have the same weight. Here, you can assume that the graph is sparse, say, $m = O(n \log n)$. For a cycle C with edges $e_1 e_2 \cdots e_\ell$, define the *max-norm* of C to be

$$f(C) = \max\{w(e_1), w(e_2), ..., w(e_\ell)\}/\ell.$$

27.A. (20 PTS.) Let f^* be the maximum max-norm of any closed walk W in the graph. Show that there is a *simple* cycle C such that $f(C) = f^*$. A simple cycle can visit a vertex at most once.

27.B. (80 PTS.) Give an algorithm that finds the cycle with largest *max-norm* as quickly as possible.