Problem Set 5

All parts are due Thursday, November 20 at 11:59PM. Please download the .zip archive for this problem set, and refer to the README.TXT file for instructions on preparing your solutions. Remember, your goal is to communicate. Full credit will be given only to a correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

Part A

Problem 5-1. [15 points] **Which truck to haul with?**

A highway network is represented as an undirected graph G with edges corresponding to roads and vertices corresponding to road intersections. Each road is labeled by a positive number describing the maximum possible height of the vehicles that can pass through the road. Given two vertices s and t, describe an efficient algorithm that computes the maximum possible height of vehicles successfully traveling from s to t. Briefly argue that your algorithm is correct and obtain its asymptotic running time.

Problem 5-2. [15 points] **Integer Weights**

Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \to \{0, 1, \ldots, W\}$ for some nonnegative integer W. Modify Dijkstra's algorithm to compute the shortest path weights from a given source vertex $s \in V$ in $O(W|V| + |E|)$ time.

For simplicity, ignore the time needed for manipulating and arithmetic operations on integers bounded by W and assume such operations take unit cost.

Problem 5-3. [10 points] **Search for diamonds**

In an undirected graph, a *diamond* is a cycle of length four. Design an algorithm that, given a graph G with n vertices, decides whether G contains a diamond in time $O(n^3)$. Argue that your algorithm is correct.

You will receive up to 5 bonus points if you find a correct algorithm that runs faster than the above time.

Problem 5-4. [20 points] **True or False.**

Decide whether these statements are **True** or **False**. You must briefly justify all your answers (that is, provide proof for True statements and counterexamples for False ones) to receive full credit.

(a) **(5 points)** If all the edge weights in the graph are multiplied by a constant positive integer a, the shortest path P between any two vertices s and t remain the same.

(b) **(5 points)** If some edge weights are negative, the shortest paths from s can be obtained by adding a constant C to every edge weight, large enough to make all edge weights nonnegative, and running Dijkstra's algorithm.

(c) **(5 points)** Let P be a shortest path from some vertex s to some other vertex t. If the weight of each edge in the graph is squared, P remains a shortest path from s to t.

(d) **(5 points)** The longest path between two vertices s and t in a directed graph with n vertices and m edges can be found in time $O(nm)$. Assume that no positive-weight cycles can be reached from s.

Part B

Problem 5-5. [40 points] **Not all avatars are created equal**

The research scientists in *Avatar 2* have built several biospheres spread across Hesiod. They need to get some supplies from one biosphere to another. Like Pandora, the atmosphere of Hesiod is poisonous to humans. As such, the scientists decide to use avatars to make the journey and carry the supplies. The biospheres are very far away from each other and it takes many days and a lot of energy for an avatar to travel from one biosphere to another. They are so far apart, in fact, that an avatar must sleep and re-energize many times during the trip. Luckily, there are stations spread throughout the world where the avatars can safely sleep and fully recover their energy. Unfortunately though, the Na'vi run these stations and, due to a slight bitterness toward avatars, charge avatars based on how much time they sleep plus a fixed fee of $2 (they use U.S. dollars for whatever reason). The research team is trying to calculate which avatars will be best for the journey, and how much it will cost each avatar to make the trip. They are completely stumped and need your help.

(a) Implement the function `travel_cost(avatar, distance, stations)` that takes in an avatar, the distance (in kilometers) to the destination biosphere, and a list of stations. This function calculates the minimum cost for the given avatar to travel to the destination biosphere, stopping at necessary stations for rest. You will be provided with `Avatar` and `Station` classes that maintain relevant information such as an avatar's total energy level, and the station's location. More details are provided in the `solution_template.py` file.

Avatars will never sleep if they are above 50% of their full energy level unless they need to in order to get to the next station. Additionally, when an avatar falls asleep, it will not wake up until it is fully re-energized. You can assume that each avatar starts the trip after a full night's sleep, or in other words with full energy. For simplicity, we will only consider the one-dimensional case where the starting biosphere is at 0 and the destination is at a positive value, with rest stations at discrete points in between. Stations will be given in sorted ascending order keyed by their position. We leave it as an exercise for you to find the best runtime possible for this problem. You are free to use any python standard functions and libraries.

Part C

Problem 5-6. [5 bonus points] **Piazza poll**

Please fill out the Piazza poll indicating how much time you have spent on this problem set. Indicate in your write up whether or not you complete the poll.