Note: Your TA probably will not cover all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. They are deliberately made long so they can serve as a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 Dijkstra's Algorithm Fails on Negative Edges

Draw a graph with five vertices or fewer, and indicate the source from which you would start Dijkstra's algorithm.

(a) Draw a graph with no negative cycles for which Dijkstra's algorithm produces the wrong answer.

(b) Draw a graph with at least two negative weight edges for which Dijkstra's algorithm produces the correct answer.

2 Dijkstra's Tiebreaking

We are given a directed graph G with positive weights on its edges. We wish to find a shortest path from s to t, and, among all shortest paths, we want the one in which the longest edge is as short as possible. How would you modify Dijkstra's algorithm to this end? Just a description of your modification is needed.

Note: if there are multiple shortest paths where the longest edge is as short as possible, outputting any of them is fine.

3 Waypoint

You are given a strongly connected directed graph $G = (V, E)$ with positive edge weights, and there is a special node $v_0 \in V$. Give an efficient algorithm that computes, for all node pairs s, t, the length of the shortest path from s to t that passes through v_0. Your algorithm should take $O(|V|^2 + |E| \log |V|)$ time.

Hint: you should only need 2 calls to Dijkstra's.

4 Shortest Path Between Sets

Given a undirected weighted graph G with non-negative edge weights $w(\cdot)$, and let $d(s, t)$ be the shortest path length from s to t. Give an efficient algorithm that takes as input two subsets of vertices S and T and outputs $\min_{s \in S, t \in T} d(s, t)$, i.e. the shortest path from any vertex in S to any vertex in T. Your algorithm should take $O(|E| \log |V|)$ time.

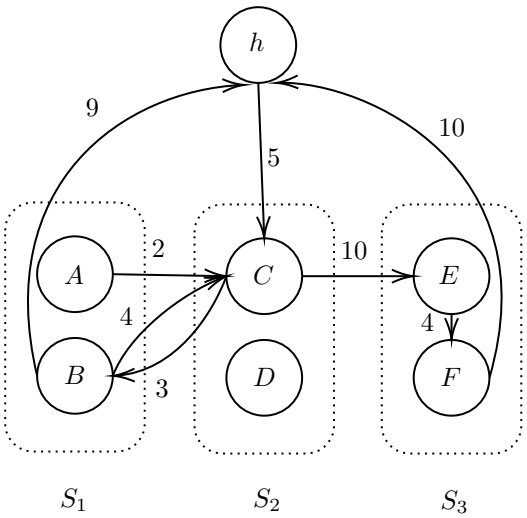
Hint: create an extra dummy node so that you can find all relevant distances by running Dijkstra's from there.

5 Running Errands

You need to run a set of k errands in Berkeley. Berkeley is represented as a directed weighted graph G, where each vertex v is a location in Berkeley, and there is an edge (u, v) with weight w_{uv} if it takes w_{uv} minutes to go from u to v. The errands must be completed in order, and we'll assume the ith errand can be completed immediately upon visiting any vertex in the set S_i (for example, if you need to buy snacks, you could do it at any grocery store). Your home in Berkeley is the vertex h.

Given G, h, and all $(S_i)_{i=1}^k$ as input, give an efficient algorithm that computes the least amount of time (in minutes) required to complete all the errands starting at h. That is, find the shortest path in G that starts at h and passes through a vertex in S_1, then a vertex in S_2, then in S_3, etc.

For instance, in the graph below, the shortest such path is $h \to C \to B \to C \to E$ and the time needed is $5 + 3 + 4 + 10 = 22$.



Hint: try creating copies of the graph G to help "keep track of" the errands you've completed so far.

6 Horn Formula Practice

(a) Find the variable assignment that solves the following horn formula:

$$(x \wedge z) \Rightarrow y, z \Rightarrow w, (y \wedge z) \Rightarrow x, \Rightarrow z, (\bar{z} \vee \bar{x}), (\bar{w} \vee \bar{y} \vee \bar{z})$$

(b) Show that any implication clause of the form $(x_i \wedge x_j \wedge \cdots) \Rightarrow \mathsf{True}$ is always satisfiable.
Hint: what disjunction clause is this equivalent to?

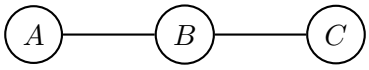
(c) Show that any implication clause of the form $\mathsf{False} \Rightarrow x_k$ is always satisfiable.

7 Graph Game

Given an undirected, unweighted graph G, with each node v having a value $\ell(v) \geq 0$, consider the following game.

1. All nodes are initially *unmarked* and your score is 0.
2. Choose an unmarked node u. Let $M(u)$ be the *marked* neighbours of u. Add $\sum_{v \in M(u)} \ell(v)$ to your score. Then mark u.
3. Repeat the last step for as many turns as you like, or until all the nodes are marked.

For instance, suppose we had the graph:



with $\ell(A) = 3$, $\ell(B) = 2$, $\ell(C) = 3$. Then, an optimal strategy is to mark A then C then B giving you a score of $0 + 0 + 6$. We can check that no other order will give us a better score.

(a) Is the following statement true or false? **For any graph, an optimal strategy which marks all the vertices always exists.** Briefly justify your answer.

(b) Give a greedy algorithm to find the order which gives the best score. **Give the algorithm description and runtime; no need to provide proof of correctness.**

(c) Now suppose that $\ell(v)$ can be negative. Give an example where your algorithm fails.

(d) Your friend suggests the following modified algorithm: delete all v with $\ell(v) < 0$, then run your greedy algorithm on the resulting graph. Give an example where this algorithm fails.

8 MST Practice (Optional)

Let $G = (V, E)$ be an undirected, connected graph.

(a) Prove that there is a unique MST if all edge weights are distinct.

(b) True or False? If G has more than $|V| - 1$ edges, and there is a unique heaviest edge, then this edge cannot be part of a MST.

(c) True or False? If the lightest edge in G is unique, then it must be a part of every MST.

9 MST Variant (Optional)

Give an undirected graph $G = (V, E \cup S)$ with edge weight $c(e)$. Note that S is disjoint with E. Design an algorithm to find a minimum one among all spanning trees having at most one edge from S and others from E.

Input: A graph $G = (V, E \cup S)$, and a cost function $c(e)$ defined for every $e \in E \cup S$.

Output: A tree $T = (V, E')$ such that T is connected (there is a path in T between any two vertices in V), $E' \subseteq E \cup S$, $\sum_{e \in E'} c(e)$ is minimized, and $|E' \cap S| \leq 1$.

Give a 3-part solution.