Problem Set 4

> *Consider a directed graph with real-valued edge lengths and no negative-cost cycles. Let s be a source vertex. Assume that there is a unique shortest path from s to every other vertex. What can you say about the subgraph of G that you get by taking the union of these shortest paths? [Pick the strongest statement that is guaranteed to be true.]*
>
> - *It has no strongly connected component with more than one vertex.*
> - *It is a path, directed away from s.*
> - *It is a tree, with all edges directed away from s.*
> - *It is a directed acyclic subgraph in which s has no incoming arcs.*

ANSWER: Option 3 is correct. Subpaths of shortest paths must themselves be shortest paths. Combining this with uniqueness, the union of shortest paths cannot include two different paths between any source and destination.

> *Consider the following optimization to the Bellman-Ford algorithm. Given a graph $G=(V,E)$ with real-valued edge lengths, we label the vertices $V=\{1,2,3,\ldots,n\}$. The source vertex s should be labeled "1", but the rest of the labeling can be arbitrary. Call an edge $(u,v)\in E$ forward if $u<v$ and backward if $u>v$. In every odd iteration of the outer loop (i.e., when $i=1,3,5,\ldots$), we visit the vertices in the order from 1 to n. In every even iteration of the outer loop (when $i=2,4,6,\ldots$), we visit the vertices in the order from n to 1. In every odd iteration, we update the value of $A[i,v]$ using only the forward edges of the form (w,v), using the most recent subproblem value for w (that from the current iteration rather than the previous one). That is, we compute $A[i,v]=\min\{A[i-1,v],\min_{(w,v)}A[i,w]+c_{wv}\}$, where the inner minimum ranges only over forward edges sticking into v (i.e., with $w<v$). Note that all relevant subproblems from the current round ($A[i,w]\forall w<v$ with $(w,v)\in E$) are available for constant-time lookup. In even iterations, we compute this same recurrence using only the backward edges (again, all relevant subproblems from the current round are available for constant-time lookup). Which of the following is true about this modified Bellman-Ford algorithm?*
>
> - *This algorithm has an asymptotically superior running time to the original Bellman-Ford algorithm.*
> - *It correctly computes shortest paths if and only if the input graph has no negative-cost cycle.*
> - *It correctly computes shortest paths if and only if the input graph is a directed acyclic graph.*
> - *It correctly computes shortest paths if and only if the input graph has no negative edges.*

Option 2 is correct. This is Yen's improvement to Bellman-Ford (https://walkccc.github.io/CLRS/Chap24/Problems/24-1/). Recall that by relaxing the edges of a weighted DAG according to a topological sort of its vertices, we can compute shortest paths from a single source in $\Theta(V+E)$ time (CLRS section 24.2). Roughly, pass i of this optimized Bellman-Ford algorithm computes shortest paths amongst those comprising at most i "alternations" between forward and backward edges.

> *Consider a directed graph in which every edge has length 1. Suppose we run the Floyd-Warshall algorithm with the following modification: instead of using the recurrence $A[i,j,k] = \min\{A[i,j,k-1], A[i,k,k-1] + A[k,j,k-1]\}$, we use the recurrence $A[i,j,k] = A[i,j,k-1] + A[i,k,k-1] * A[k,j,k-1]$. For the base case, set $A[i,j,0] = 1$ if (i,j) is an edge and 0 otherwise. What does this modified algorithm compute – specifically, what is $A[i,j,n]$ at the conclusion of the algorithm?*
>
> - *The length of a longest path from i to j.*
> - *The number of simple (i.e., cycle-free) paths from i to j.*
> - *The number of shortest paths from i to j.*
> - *None of the other answers are correct.*

ANSWER: Option 1 is incorrect, nothing in the recurrence is indicative of the longest path from i to j (you'd expect a max somewhere).

Option 2 is incorrect, nothing in the recurrence prevents cycles.

Option 1 is incorrect, nothing in the recurrence is indicative of the shortest path from i to j (you'd expect a min somewhere).

Option 4 is correct. The recurrence computes the number of paths from from i to j, including those containing cycles.

> *Suppose we run the Floyd-Warshall algorithm on a directed graph $G = (V, E)$ in which every edge's length is either -1, 0, or 1. Suppose further that G is strongly connected, with at least one $u - v$ path for every pair u, v of vertices. The graph may or may not have a negative-cost cycle. How large can the final entries $A[i,j,n]$ be, in absolute value? Choose the smallest number that is guaranteed to be a valid upper bound. (As usual, n denotes $|V|$.) [WARNING: for this question, make sure you refer to the implementation of the Floyd-Wardshall algorithm given in lecture, rather than to some alternative source.]*
>
> - n^2
> - $+\infty$
> - $n - 1$
> - 2^n

ANSWER: Lets prove by induction that $A[i, j, k] = -2^k$. Lets further assume that all edge lengths are -1. From the lecture, $A[i, j, 0] = C_{ij} \forall i \neq j$, which by the assumption that all edge lengths are -1, is -2^0. Thus, the base case is proved.

Now consider the recurrence $A[i, j, k] = \min\{A[i, j, k-1], A[i, k, k-1] + A[k, j, k-1]\} \forall k > 0$. For $k+1$, the R.H.S. is equal to $\min\{-2^k, -2^k - 2^k\}$, or -2^{k+1}. Thus, it is true in general for all values of k, and since the maximum value of k can be n, option 4 is correct.

Note that the induction doesn't hold if we consider all edge lengths equal to 1, since the minimum is going to be the first term, not the sum of two terms.

> *Which of the following events cannot possibly occur during the reweighting step of Johnson's algorithm for the all-pairs shortest-paths problem? (Assume that the input graph has no negative-cost cycles.)*
>
> - *Reweighting strictly increases the length of some $s - t$ path, while strictly decreasing the length of some $t - s$ path.*
> - *The length of some edge strictly decreases after the reweighting.*
> - *In a directed acyclic graph, reweighting causes the length of every path to strictly increase.*
> - *In a directed graph with at least one cycle, reweighting causes the length of every path to strictly increase.*

ANSWER: Reweighting makes all path lengths non-negative, but it also reduces the length for some. Thus, option 1 is incorrect.

Option 2 and 3 are incorrect by the same argument.

Consider two "halves" of a cycle: The increase in length of one of these paths equals the decrease in length of the other path. Thus, option 4 is correct (it cannot possibly occur during the reweighting step).

> *Consider a directed graph with real-valued edge lengths and no negative-cost cycles. Let s be a source vertex. Assume that each shortest path from s to another vertex has at most k edges. How fast can you solve the single-source shortest path problem? (As usual, n and m denote the number of vertices and edges, respectively.) [Pick the strongest statement that is guaranteed to be true.]*
>
> - $O(kn)$
> - $O(km)$
> - $O(mn)$
> - $O(m+n)$

ANSWER: Option 1 is correct. We can limit the number of edges in the Bellman-Ford algorithm to k.