Problem Set #1

Quiz, 5 questions

5/5 points (100%)

Congratulations! You passed!

Next Item

1/1 point

1. 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 is a path, directed away from s.
- It is a directed acyclic subgraph in which s has no incoming arcs.
- It is a tree, with all edges directed away from s.

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.

- It has no strongly connected component with more than one vertex.

1 / 1 point

2. 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 ($A[i, w]$ for all $w < v$ 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?

- ( ) 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.
- ( ) This algorithm has an asymptotically superior running time to the original Bellman-Ford algorithm.
- (x) It correctly computes shortest paths if and only if the input graph has no negative-cost cycle.

Correct

Indeed. Can you prove it? As a preliminary step, prove that with a directed acyclic graph, considering destinations in topological order allows one to compute correct shortest paths in one pass (and thus, in linear time). Roughly, pass i of this optimized Bellman-Ford algorithm computes shortest paths amongst those comprising at most i "alternations" between forward and backward edges.

1 / 1
point

3. 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.]

- $O(mn)$
- $O(km)$

Correct
Right, you can stop the Bellman-Ford algorithm after k iterations

- $O(kn)$
- $O(m+n)$

1 / 1
point

4. 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 number of shortest paths from i to j.
- The number of simple (i.e., cycle-free) paths from i to j.
- The length of a longest path from i to j.
- None of the other answers are correct.

Correct
Indeed. How would you describe what the recurrence is in fact computing?

1 / 1 point

5. 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 G 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-Warshall algorithm given in lecture, rather than to some alternative source.]

- n^2
- $+\infty$
- 2^n

Correct
By induction. Can you prove a sharper (exponential) bound, or is this tight?

- $n - 1$