

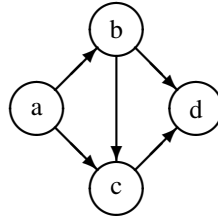
COMP4500/7500 Advanced Algorithms & Data Structures

Tutorial Exercises 5 (2014/2)*

School of Information Technology and Electrical Engineering, University of Queensland

August 27, 2014

1. The following is a small example of a directed acyclic graph (dag).



In this dag there is only one topological ordering: $abcd$. There are three paths from a to d : abd , $abcd$, and acd .

A directed acyclic graph (dag) may have many distinct topological orderings. Give a dag with n nodes which has the minimum possible number of topological orderings? Which dag has the maximum number?

2. Design an abstract algorithm to list all the topological orderings of a given directed acyclic graph. You may use the abstract traversals of graphs, and make use of abstract set and sequence operations, e.g., set union and subtraction, and sequence concatenation. You may also assume the procedure to calculate the in-degrees of a graph (from Revision 4) is available. What are the worst-case time and space complexities of your algorithm?
3. Find a recurrence equation which computes the number of paths between two vertices of a directed acyclic graph.
4. (CLRS Exercise 23.1-7; CLR Exercise 24.1-7)

Argue that if all edge weights of a graph are positive, then any subset of edges that connects all vertices and has minimum total weight must be a tree. Give an example to show that the same conclusion does not hold if we allow some weights to be nonpositive.

5. The Bellman-Ford algorithm solves the single-source shortest-paths problem on a weighted, directed graph $G = (V, E)$ as long as there are no negative weight cycles. The algorithm returns TRUE if and only if the graph contains no negative weight cycles that are reachable from the source. In that case the final shortest-path weight from the source s to a vertex v is stored in $v.d$ at the termination of the algorithm. For each vertex v , on termination $v.\pi$ stores the predecessor on the path from s to v . The following pseudo-code describes an implementation of the Bellman-Ford algorithm. These procedures are taken from CLRS p648–651 [3rd].

BELLMAN-FORD(G, w, s)

```
1 //  $G$  is the graph,  $w$  the weight function,  $s$  the source vertex
2 INIT-SINGLE-SOURCE( $G, s$ )
3 for  $i = 1$  to  $|G.V| - 1$ 
4   for each edge  $(u, v) \in G.E$ 
5     RELAX( $u, v, w$ )
6 for each edge  $(u, v) \in G.E$ 
7   if  $v.d > u.d + w(u, v)$ 
8     return FALSE
9 return TRUE
```

*Copyright © reserved August 27, 2014

INIT-SINGLE-SOURCE(G, s)

```

1  for each vertex  $v \in G.V$ 
2       $v.d = \infty$ 
3       $v.\pi = \text{NIL}$ 
4   $s.d = 0$ 

```

RELAX(u, v, w)

```

1  if  $v.d > u.d + w(u, v)$ 
2       $v.d = u.d + w(u, v)$ 
3       $v.\pi = u$ 

```

Consider the graph on vertices v_1, v_2, v_3, v_4, v_5 described by the following adjacency matrix w :

	v_1	v_2	v_3	v_4	v_5
v_1	∞	5	8	-4	∞
v_2	-2	∞	∞	∞	∞
v_3	∞	-3	∞	9	∞
v_4	∞	7	∞	∞	2
v_5	6	∞	7	∞	∞

For vertices u and v , an entry x in row u and column v of the adjacency matrix means that the distance from u to v on the directed edge (u, v) is x , that is, $w(u, v) = x$. If the distance is shown as ∞ then u is not connected to v in that direction.

- Draw a pictorial representation of the graph, showing vertices, edges and distances.
- Show how the Bellman-Ford algorithm runs on this directed graph, using vertex v_5 as the source. Make the following assumptions about the implementation. An adjacency list representation is used in such a way that edges are made available in lexicographic order in the loops:
“**for** each edge $(u, v) \in G.E$ ”.
This means that (v_i, v_j) precedes (v_k, v_ℓ) if $i < k$ or if $i = k$ and $j < \ell$.
Give your answer by showing the values of $v.d$ and $v.\pi$ (as used in the pseudocode) for each vertex v after each iteration of the first **for** loop in the BELLMAN-FORD procedure.
- What is the return value of the BELLMAN-FORD procedure in this case (TRUE or FALSE)?
- Analyse the complexity of the Bellman-Ford algorithm considering the most appropriate graph representation. Give your answer using O -notation in terms of $|V|$ and $|E|$, where these represent the numbers of vertices and edges, respectively. Make your bound as tight as possible. Provide a justification of your analysis.

6. (CLR Exercise 25.1-4)

Let $G = (V, E)$ be a weighted, directed graph with source vertex s , and let G be initialized by the above procedure INIT-SINGLE-SOURCE(G, s). Prove that if a sequence of relaxation steps sets $s.\pi$ to a non-NIL value, then G contains a negative-weight cycle.

7. Design an algorithm to compute single-source shortest paths in a directed acyclic graph in $O(|V| + |E|)$. (Hint: start by doing a topological sort.)

8. (CLRS Exercise 24.1-3; CLR Exercise 25.3-3)

Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all pairs of vertices $u, v \in V$ of the minimum number of edges in a shortest path from u to v . (Here, the shortest path is by weight, not the number of edges.) Suggest a simple modification to the Bellman-Ford algorithm that allows it to terminate in $m + 1$ passes.