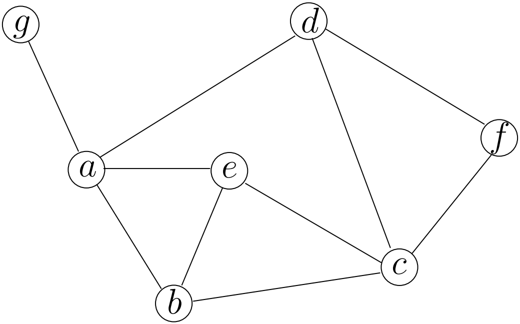
COMP3711: Design and Analysis of Algorithms

Tutorial 9

HKUST

The adjacency list representation of a graph G, which has 7 vertices and 10 edges, is:

$$
\begin{array}{ll}
a :\rightarrow d, e, b, g & b :\rightarrow e, c, a \\
c :\rightarrow f, e, b, d & d :\rightarrow c, a, f \\
e :\rightarrow a, c, b & f :\rightarrow d, c \\
g :\rightarrow a &
\end{array}
$$



Question 1

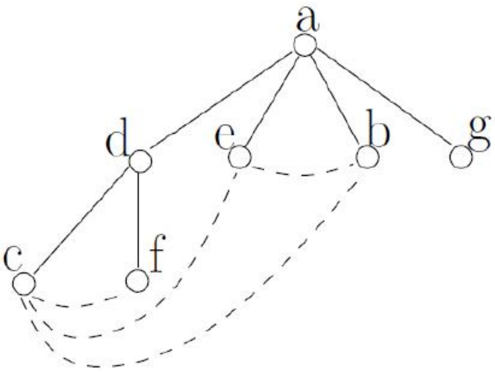
(a) Show the breadth-first search tree by running BFS on graph G with the given adjacency list, use vertex a as the source.

(b) Show the edges which are not presented in the BFS tree in part (a) by dashed lines.

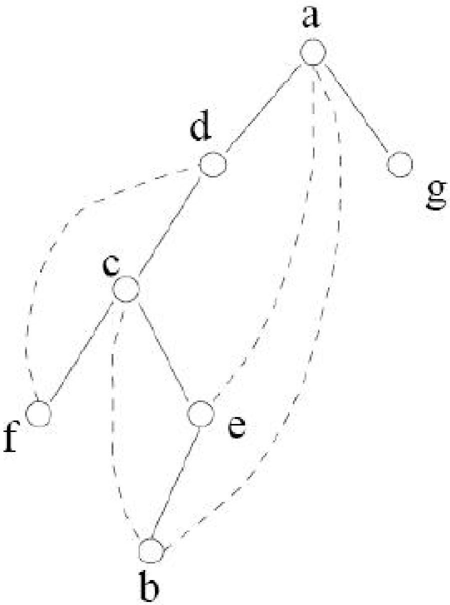
(c) Show the depth-first search tree by running DFS on graph G with the given adjacency list, use vertex a as the source.

(d) Show the edges which are not presented in the DFS tree in part (c) by dashed lines.

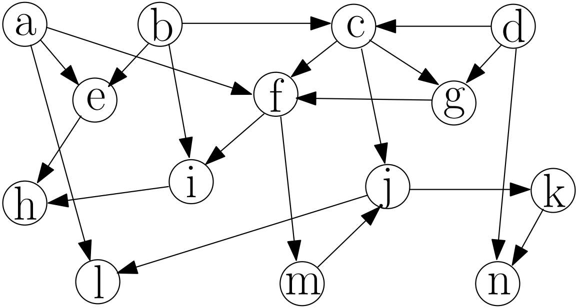
Solution 1 a & b



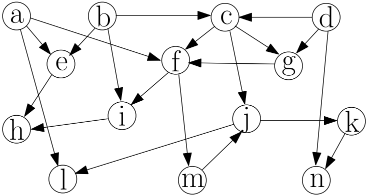
Solution 1 c & d



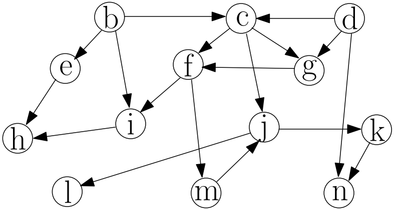
Show the topological ordering of the following graph.



Solution 2



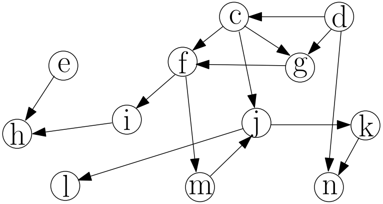
$Q = \{a, b, d\}$



$Q = \{b, d\}$

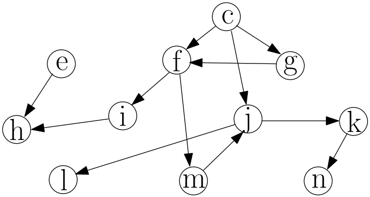
Output: a

Solution 2



$Q = \{d, e\}$

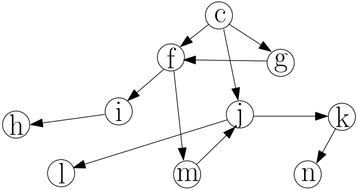
Output: a, b



$Q = \{e, c\}$

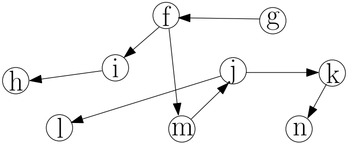
Output: a, b, d

Solution 2



$Q = \{c\}$

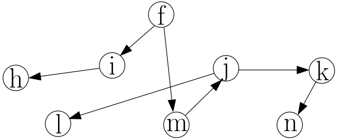
Output: a, b, d, e



$Q = \{g\}$

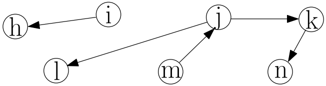
Output: a, b, d, e, c

Solution 2



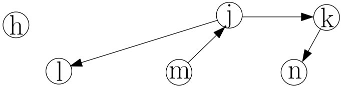
$Q = \{f\}$

Output: a, b, d, e, c, g



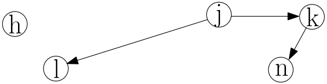
$Q = \{i, m\}$

Output: a, b, d, e, c, g, f



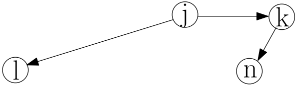
$Q = \{m, h\}$

Output: a, b, d, e, c, g, f, i



$Q = \{h, j\}$

Output: $a, b, d, e, c, g, f, i, m$



$Q = \{j\}$

Output: $a, b, d, e, c, g, f, i, m, h$

$Q = \{k, l\}$

Output: $a, b, d, e, c, g, f, i, m, h, j$

$Q = \{l, n\}$

Output: $a, b, d, e, c, g, f, i, m, h, j, k$

$Q = \{n\}$

Output: $a, b, d, e, c, g, f, i, m, h, j, k, l$

$Q = \{\}$

Output: $a, b, d, e, c, g, f, i, m, h, j, k, l, n$

Question 3

Given an undirected weighted graph $G = (V, E)$ with non-negative distinct edge weight and an MST T of it. (a) Replace the weight of each edge w by w^2. Is T still an MST for the new graph? (b) Next we consider a shortest path $u \rightarrow v$ in the original graph. Is this path still a shortest path from u to v in the new graph? If yes, prove so; if not, give a counter example.

Solution 3

(a) Yes, the MST remains unchanged when the weights are changing from w to w^2. There are two ways to prove it:

(1) We observe that Prim's algorithm only compares the weights of edges, and changing the weights from w to w^2 does not change the result of any comparison. Thus Prim's algorithm's output will remain the same. From the correctness of Prim's algorithm, we conclude that T is still the MST of the new graph.

(2) We can mimic the proof on the uniqueness of MST. Consider any edge e of T. Removing e from T breaks it into S and $V - S$. e is the min cost edge crossing the cut in the original graph, so it must still be the min cost edge crossing the cut in the modified graph. Apply the cut lemma on S, and we know that any MST in the new graph must contain e. This argument holds for every edge e of T, thus T remains the MST of the new graph.

Solution 3

(b) No, it may not be a shortest path in the new graph.

Example