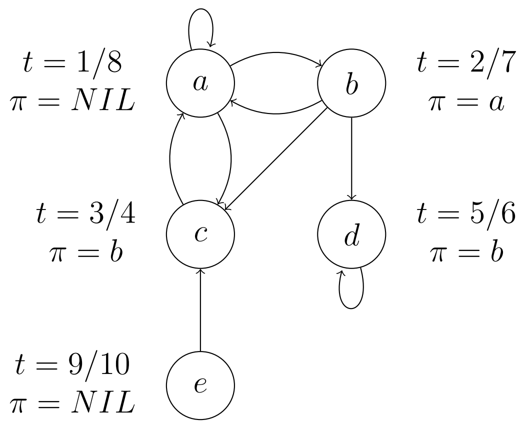
Data Structure and Algorithm Analysis(H)
Southern University of Science and Technology
Mengxuan Wu
12212006

Work Sheet 14

Mengxuan Wu

Question 14.1

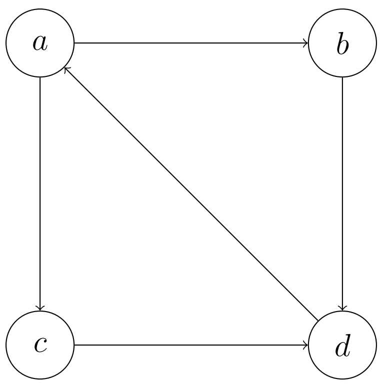


Question 14.2

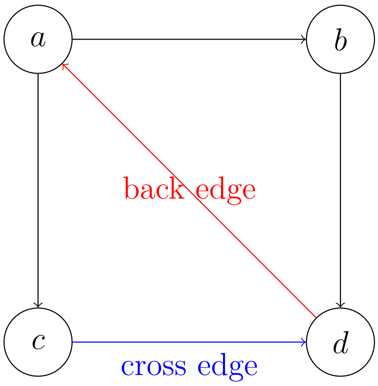
Disproof.

No. The number of back edges might change because one back edge can participate in multiple cycles.

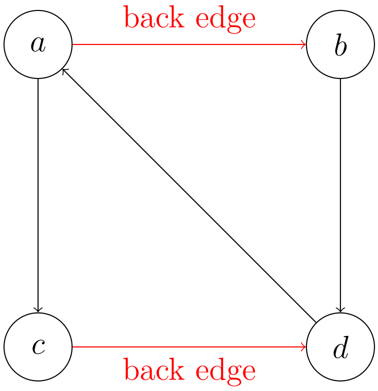
Consider the following graph:



If we run DFS on vertex a, we will get the following DFS tree:

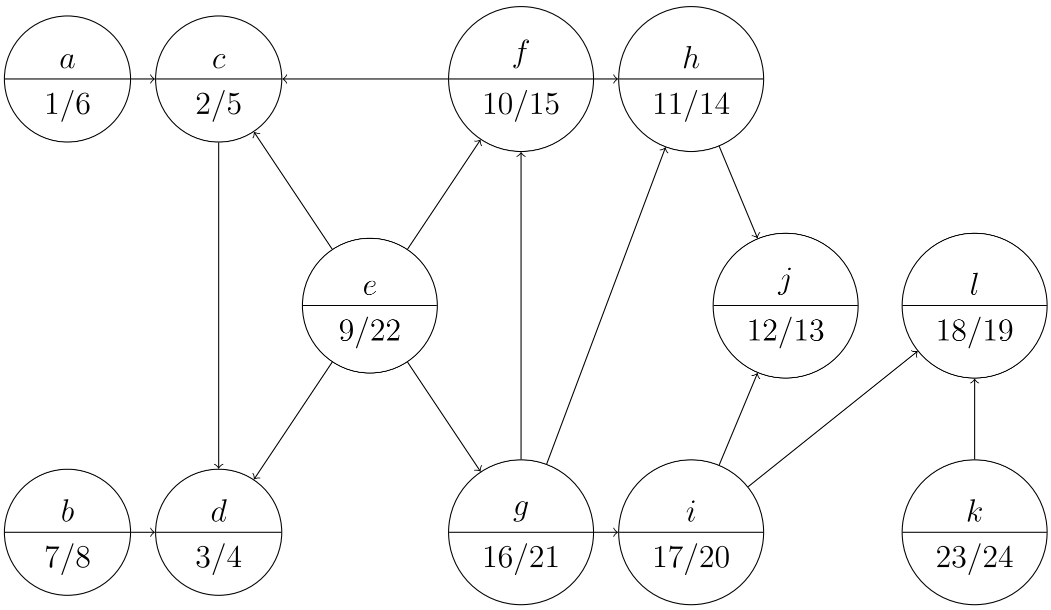


If we run DFS on vertex b, we will get the following DFS tree:



□

Question 14.3



The list sorted by finishing time is: $k, e, g, i, l, f, h, j, b, a, c, d$.

Question 14.4

Check-Is-Acyclic($G(V, E)$)

if $|E| > |V| - 1$
    return False
DFS'($G(V, E)$)
for each edge $(u, v) \in E$
    if (u, v) is a back edge
        return False
return True

For an undirected graph, if there is no cycle in the graph, then it must be a tree or a forest. And for a tree or a forest, $|E| \leq |V| - 1$ must be true. Hence, we check this condition first. If the condition is not satisfied, we return False immediately and runtime is $O(1)$.

If the condition is satisfied, we run DFS on the graph. Here we can modify the DFS algorithm to explicitly mark each back edge, since it only takes $O(1)$ time, the runtime of DFS is still $O(|V| + |E|)$. Then, we check each edge in the graph.

The runtime of DFS is $O(|V| + |E|)$. Since we know $|E| \leq |V| - 1$, we can infer that $O(|V| + |E|) = O(|V|)$. Then, the total runtime of the algorithm is $O(|V|) + O(|V|) = O(|V|)$.

Hence, in any case, the runtime of the algorithm is $O(|V|)$.