Course: Design and Analysis of Algorithms

Worked out Assignment on Unique Path Graphs

An Important guideline

- It is only through the assignments that one learns the most about the algorithms and data structures. For the current assignment, a sequence of hints are given. Try to look at a hint only after trying on your own for sufficiently long period of time. The onus of learning from a course lies first on you. So act wisely while working on this assignment.

An efficient algorithm for the unique-paths problem

Let $G = (V, E)$ be a directed graph on $n = |V|$ vertices and $m = |E|$ edges. As stated in one of the lectures in the course, G is said to be a unique path graph if the following condition holds for each pair of vertices $u, v \in V$.

There is **at most 1** *path from* u *to* v.

There is an easy $\mathcal{O}(mn)$ time algorithm for determining if G is a unique-path graph. This algorithm just uses the **classification of non-tree edges** in a DFS traversal which we discussed thoroughly in the lecture. Make sincere attempts to design this algorithm.

Go to the next page only after you have spent sufficient time on your own.

Hint 1 : Just look at the time complexity $\mathcal{O}(mn)$ that we are aiming. It suffices if we can design an $\mathcal{O}(m)$ time algorithm that, for a given $u \in$V, can determine if there is at most 1 path from u to every other vertex.

Make use of the above mentioned hint and try to design an $\mathcal{O}(m)$ time algorithm for this subproblem.

Go to the next page only after you have spent sufficient time on your own.

Hint 2 : Perform DFS traversal from u. Let T_u be the DFS tree thus obtained. This will classify all the non-tree edges into forward edges, cross edges, and backward edges.

Study the following questions carefully.

1. Does presence of a forward edge imply that there are multiple paths from u ?
2. Does presence of a cross edge imply that there are multiple paths from u ?
3. Does presence of a backward edge imply that there are multiple paths from u ?

Try to find the answers to these questions earnestly ...

Go to the next page only after you have spent sufficient time on your own.

The answers to the questions asked on the previous page are as follows.

1. Presence of a forward edge indeed implies that there are multiple paths from u.

2. Presence of a cross edge indeed implies that there are multiple paths from u.

So if you encounter any cross edge or a forward edge, we may stop. So we assume without loss of generality that there is no forward edge or a cross edge. In this scenario, does the presence of any backward edge imply multiple paths **originating from** u ? The answer is no.

This gives an $\mathcal{O}(m)$ time algorithm to determine of there is at most 1 path from u to any other vertex in G. Repeating this algorithm for each vertex in V (carry out a fresh DFS traversal from each vertex) gives an $\mathcal{O}(mn)$ time algorithm for the problem.

Those whose interest is just to score a good grade in the course may stop at this point. Others (hopefully nonzero number of students) may go to the next page. :-)

The aim is to design an $O(n^2)$ time algorithm for this problem. The biggest hint is that you need to carefully look at the $O(mn)$ time algorithm. This is because the (n^2) time algorithm is a *refined* version of this algorithm only. So try for sometime on your own, sincerely, patiently, and persistently.

If you still do not succeed, do not feel discouraged. Go to the following page for more explicit hints.

In the $O(mn)$ time algorithm, we are doing n DFS traversals - one from each vertex. The aim of doing DFS traversal from a vertex, say u, is to determine whether there are multiple paths from u to any other vertex. Recall that if we encounter any forward edge or cross edge, we stop since that would imply that there are multiple paths from u. However, we totally *discard* the backward edges. This was because no backward edge during DFS(u) can be part of any path from u. However, observe that these backward edges may potentially lead to multiple paths from a vertex other than u to some other vertex. Ponder over it before going to the next page.

Consider the DFS tree obtained by carrying out DFS traversal of G from a vertex, say u. Let us denote it by T_u. What if there is a vertex which has more than one backward edges emerging from a vertex, say v ? What can you infer in this case ?

Go to the next page after spending sufficient time finding answer to these questions.

The answer to the question stated in the previous page is the following: If there are multiple backward edges from v in T_u, then there are multiple paths from v to at least one vertex; and this vertex is the lower of the 2 ancestors to which v is having backward edges.

So the following question is the last hint. Make use of the hint to design the algorithm

Hint : How many backward edges can you afford to encounter during a fresh DFS traversal from a vertex u ?