
Problem 7.21. Let G represent an undirected graph. Also let

$$SPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b\},$$

and

$$LPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at least } k \text{ from } a \text{ to } b\}.$$

Part a. Show that $SPATH \in P$.

Proof. We show that $SPATH \in P$ by presenting a polynomial time algorithm that decides $SPATH$. A polynomial time algorithm M for $SPATH$ operates as follows.

$M =$ "On input $\langle G, a, b, k\rangle$:

1. Unmark all nodes.
2. Assign each node a value of ∞.
3. Mark a, and set it's value 0.
4. Let integer $d = 0$.
5. Repeat until no additional nodes are marked:
6. $\quad d = d + 1$.
7. $\quad$ Scan all edges of G. If an edge (u, v) exists between a marked node u and an unmarked node v, then mark v and set value of v to d.
8. If b is marked and value of b is at most k, then *Accept*. Otherwise *reject*."

Now we analyze this algorithm to show that it runs in polynomial time. Obviously, stages 1 to 4 and stage 8 are executed only once. e. Stage 7 runs at most m times because each time except the last it marks an additional node in G, where m is the number of nodes in G. Stage 6 also runs at most m times. Thus, the total number of stages used is at most $1 + 1 + m + m$, giving a polynomial in the size of G.

Stages 1 to 4 along with stages 6 and 8 are easily implemented in polynomial time on any reasonable deterministic model. Stage 7 involves a scan of the input and a test of whether certain nodes are marked and an update of node values, which also is easily implemented in polynomial time. Hence M is a polynomial time algorithm for $SPATH$. □

Part b. Show that $LPATH$ is NP-complete.

Proof. □