

CS217 – Algorithm Design and Analysis

Homework 3

Not Strong Enough

April 19, 2020

1

Suppose the edges e_1, \dots, e_m are sorted by their cost. Show how to solve MCP in time $O(n + m)$

Solution. The algorithm is executed in iterations. Suppose that $c(e_1) \geq c(e_2) \geq \dots \geq c(e_m)$. In iteration i , consider e_i .

There is a set of reachable vertices, and the initial status of the set is s . Each vertex has an predecessor recording the predecessor in the path by which s reaches the vertex. Its initial status is *null*. Each vertex also has a set of unresolved edges whose start point is the vertex. Its initial status is \emptyset

In each iteration, denote the handled edge by $e = (u, v)$.

If u is not reachable, simply add e into the unresolved edge set of u .

If u, v are already reachable, nothing needs to be done.

If u is already reachable but v is not, set v reachable and set its precursor be u . Then handle all resolved edges of v . (let the target of each edge be reachable if it is not) Since this may introduce new reachable vertices, handle them too. This procedure is like a BFS.

Once t is reached, the algorithm terminates, and the cost of the newly introduced edge is the cost of MCP. Using the predecessor of each vertex, the path can be generated. \square

First prove that each vertex in the reachable set is really reachable(so it is well-defined):

Proof. The reachable set has the property that, except for s , all vertices in the set have a predecessor also in the set.

This is trivial since when we add an vertex to the reachable set, we set its precursor by a reachable vertex.

Using this property, since its precursor is reachable, the vertex itself is reachable as well. \square

Then prove that all reachable vertices using $\{e_1, \dots, e_n\}$ are in reachable set after iteration n :

Proof. Assume that there is a path from s to v using edges $e_{i_1}, e_{i_2} \dots e_{i_k}, (i_1, i_2 \dots i_k \in \{1, 2, \dots, n\})$. And $\forall j \in \{1, 2 \dots k\}, e_{i_j} = (u_j, u_{j+1}); u_1 = s, u_{k+1} = v$. Let $M_j = \max\{i_1, i_2 \dots i_j\}, m_j = \min\{i_1, i_2 \dots i_j\}$

Assume that after iteration M_{q-1} $u_1, u_2 \dots u_q$ are in reachable set and u_q is added in iteration M_{q-1} . If $i_q < M_{q-1}$, e_q is in unresolved set of u_q . So in iteration $M_{q-1} = M_q$, when adding u_q in reachable set, all unresolved edges of u_q are handled so u_{q+1} is added as well. If $i_q > M_{q-1}$, in iteration $M_q = i_q$, u_q is in reachable set so u_{q+1} is added in reachable set.

Consider the initial status that after iteration $M_1 = i_1$, $u_1 = s, u_2$ are in reachable set and u_2 is added in iteration M_1 .

By induction, it is proved that after iteration M_k , $u_1, u_2 \dots u_{k+1}$ are all in reachable set. \square

Now it is trivial to prove the correctness of the algorithm:

If the cost of MCP is $c = e_n$, then the path only use edges in $\{e_1, \dots, e_n\}$, so exactly in iteration n , t is reachable.

Then prove the complexity of the algorithm is $O(n + m)$:

Proof. Consider an arbitrary edge $e_n = (u, v)$. If in iteration n , u is already reachable, it is not added into the unresolved set so it will not be visited. Otherwise, it is added into the unresolved set of u , and when u is added into the reachable set, it will be visited.

Since u can only be added to the reachable set once, e_n is visited at most twice. So the complexity is $O(m)$

In the initial step, each vertex is initialized, so the complexity is $O(n)$.

Combine the two steps together, the complexity is $O(n + m)$. □