

## Exercise 1

Suppose that  $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$ , the algorithm is described as follows.

1. Initially  $S = \{s\}, Q = \{\}, C_i = \{\}$ .
2. Traverse all edges in reversed order until  $t \in S$ , goto step 8.
3. For  $e_i = (u_i, v_i)$ , if  $u_i, v_i \in S$ , ignore this edge; if  $u_i \in S, v_i \notin S$ , let  $Q \leftarrow Q \cup \{v_i\}$ ; otherwise let  $C_{u_i} \leftarrow C_{u_i} \cup \{v_i\}$ .
4. If  $Q \neq \emptyset$ , pop a vertex  $u \in Q$ , otherwise goto step 2.
5. Let  $S \leftarrow S \cup \{u\}$ .
6. For all  $v \in C_u$ , if  $v \notin S \cup Q$ , let  $Q \leftarrow Q \cup \{v\}$ .
7. Goto step 4.
8. The weight of the edge that last visited is the answer.

Every edge is visited at most once, so the total time complexity of the algorithm is  $O(n + m)$ .

## Exercise 2

Split  $m$  edges of  $E$  into  $\log m$  parts  $E_1, \dots, E_{\log m}$  such that for any pairs of edges  $e, e'$  if  $e \in E_i, e' \in E_j$  and  $i < j$  we have  $c(e) < c(e')$ .

These parts can be obtained by the median-of-medians algorithm with time complexity of  $O(m \log \log m)$ .

Then rewrite the weights of each edge by the number of the parts, thus the edges are well sorted, then use the algorithm above. The time complexity is also  $O(m \log \log m)$ .

The overall time complexity is  $O(m \log \log m)$ .

## Exercise 3

If we divide the edges to  $\log \log \log m$  or  $\log \log \log \log m$  parts, we can get an algorithm of time complexity of  $O(m \log \log \log m)$  or  $O(m \log \log \log \log m)$

similarly.

Since  $m$  is finite, we can't apply indefinite log on  $m$ .

Let  $\log^*(n) = \min\{i \geq 0 : \log^{(i)} n \leq 1\}$ .

The best time complexity we can get is  $O(m \log^*(m))$ .

## Exercise 4

For a graph  $G$ , let  $f$  be the flow of maximum capacity path of  $G$ , and  $G_f$  be the residual network.

We have

$$c_G^* > c_{G_f}^*$$

So there are up to  $m$  different  $c_{G'}^*$ .

The other part is the same case as the proof of Edmonds-Karp algorithm.

The overall time complexity is bounded by  $O(m^3)$ .