1

Give an algorithm for MCP of running time $O(m \log\log m)$.

Solution. **Require:** edges $e_1, \cdots, e_m$

function MCP
  $C \leftarrow \{e_1, \cdots, e_m\}$
  $d \leftarrow 0$
  while $|C| > 1$ **do**
    $divisions \leftarrow C$
    for $i = 1$ to 2^d **do**
      $\{S_1, \cdots, S_k\} \leftarrow divisions$
      $m_1, \cdots, m_k \leftarrow$ the median of $S_1, \cdots, S_k$ in means of weight
      $divisions \leftarrow \bigcup_{j=1}^{k} \{s \in S_j : c(s) <= m_j\} \cup \{s \in S_j : c(s) > m_j\}$
    end for
    $C \leftarrow GetDivision(divisions)$.
    $d \leftarrow 2^d + d$
  end while
  return the only element in C
end function
function GetDivision(divisions)
  $G \leftarrow (V, \{\})$
  for div in $divisions$ **do** (iterate from higher capacity to lower capacity)
    add all edges in div to G
    if G is s-t-connected **then**
      return div
    end if
  end for
end function

The correctness of the algorithm comes from below. First, we initialize our $c*$ candidate to be all the edges in G. Then, in each iteration, we divide it into many divisions of approximately the same size in order, and test which division is $c*$ in. Finally, there will be only one candidate left, and that is $c*$.

Now we'll analyze the running time. Note that finding the medium of a set S has running time $O(|S|)$. In the inner "for" loop, we find the medium of $S_1, \cdots, S_k$ in d iterations. However there's an invariant equation $\sum_{i=1}^{k} |S_k| = |C|$. So the finding the median requires $O(|C| * 2^d)$ in total. The $GetDivision$ function needs to add all the edges into the graph in the worst case . Testing the connectivity will use $O(m)$ time in total if we record a set of reachable vertices as decribed in Problem 1. So its running time is $O(m)$ for $GetDivision$.

Observe that $|C| * 2^d = O(m)$. We'll prove it by induction. Initially, $|C| = m, d = 0$. It obviously holds. Suppose $|C_{old}| * 2^{d_{old}} = O(m)$. We have$|C_{new}| = O(|C_{old}| * \frac{1}{2}^{2^{d_{old}}})$. $d_{new} = d_{old} + 2^{d_{old}}$. So $|C_{new}| * 2^{d_{new}} = O(m)$

Now we get the running time of one "while" loop is $O(m)$. Note that $2^d < m$. Since d increases exponentially, the "while" loop will iterate for $O(\log *m)$ times. Finally, we get the running time is $O(m \log *m)$. Obviously this algorithm has a better running time than $O(m \log\log m), O(m \log\log\log m), O(m \log\log\log\log m)$ □

2

Give an algorithm for MCP that runs in $O(m \log\log\log m)$? How about $O(m \log\log\log\log m)$? How far can you get?

Solution. See Problem 2. $\square$