

Problem Set 3: Minimum Spanning Tree

1. (EASY) If all the edges in the graph have equal weights, then show that you can find the minimum spanning tree in $O(m + n)$ time.
2. (EASY) In the heart of New Delhi, esteemed mathematician Professor Gupta introduces a novel approach for computing minimum spanning trees, inspired by India's rich mathematical heritage. His algorithm unfolds as follows: Given a graph $G = (V, E)$, partition the vertex set V into two sets V_1 and V_2 such that $|V_1|$ and $|V_2|$ differ by at most 1. Let E_1 be the set of edges incident solely on vertices in V_1 , and E_2 be the set of edges incident solely on vertices in V_2 . Recursively solve the minimum spanning tree problem on the subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge in E that crosses the cut V_1, V_2 , and use this edge to unite the resulting two minimum spanning trees into a single spanning tree. Evaluate whether Professor Gupta's algorithm correctly computes a minimum spanning tree of G , or provide a counterexample to demonstrate its limitations.

Either argue that the algorithm correctly computes a minimum spanning tree of G , or provide an example for which the algorithm fails.

3. (MEDIUM) You are given a graph G and its minimum spanning tree \mathcal{M} . Design an efficient algorithm that updates \mathcal{M} when:
 - (a) A new edge e is added to the graph.
 - (b) An edge e is deleted from the graph.
4. (MEDIUM) Show that if all the weight in the graph are distinct, then there is a unique minimum spanning tree in the graph.
5. (MEDIUM) You are given a graph G and its minimum spanning tree \mathcal{M} . Design an algorithm to find if \mathcal{M} is the unique \mathcal{M} in G in $O(mn)$ time. Now assume that you have an access to an oracle that can tell you the maximum weight edge on the path between any two vertices in \mathcal{M} in $O(\log n)$ time. Now show that you can find \mathcal{M} is unique (or not) in $O(m \log n)$ time.
6. (MEDIUM) Let e be the edge in the graph with maximum weight that lies in a cycle. That is, all the other edges have weight strictly less than $w(e)$ in the graph. Show that e cannot lie in the minimum spanning tree of G .
7. (HARD) Show that for any graph G with minimum spanning trees T and T' , that for each weight w , T and T' contain the same number of edges of weight w .
8. (HARD) In the class, we did Prim's and Kruskal's algorithm. But there is another algorithm that predates both of them – Boruvka's algorithm. We will assume that all the edges have distinct weight in the graph G

```
Initially there are  $n$  singletons in the graph;
 $\mathcal{M} \leftarrow \emptyset$ ;
while  $\mathcal{M}$  is not a tree on  $n$  vertices do
  | foreach tree  $T$  in  $\mathcal{M}$  do
  | | let  $e$  be the minimum weight edge in  $G$  in the cut  $(T, V - T)$ ;
  | | Add  $e$  to  $\mathcal{M}$ ;
  | end
end
```

Prove that the above algorithm indeed returns a minimum spanning tree. Furthermore, show that Bouruvka's algorithm can be implemented in $O(m \log n)$.

[You will get a lot of satisfaction if you are able to do this question on your own without any hints. I am not going to ask this question either in exams or in any assignment.]

9. (HARD) You are given a graph G and an edge $e \in G$. You want to find if e is in the minimum spanning tree of G or not. Design an algorithm that performs this task in $O(m)$ time.
10. (HARD) All edges of the graph G can be partitioned into k classes. The cost of each edge class is different. You can purchase all edges of class i by paying c_i rupees. Your task is to purchase the minimum number of edge classes that contain the minimum spanning tree of G . Design an algorithm that accomplishes this task in $O(m \log n)$ time.
11. (HARD) In the country of Mallaga, there are n cities and m roads of three types: red, blue, and green. Men can travel on red and green roads, while women can travel on blue and green roads. To minimize maintenance costs, the ruler wants to destroy some roads while ensuring the country remains connected for both men and women. What is the maximum number of roads that can be destroyed? Design an efficient algorithm for this problem.
12. (HARD) A graph is a nearly a tree if it has $n + k$ edges where k is a small constant. Show that you can find a minimum spanning tree in such a graph in $O(nk)$ time.
13. (HARD) Design an efficient algorithm to find a spanning tree for a connected, weighted, undirected graph such that the weight of the maximum weight edge in the spanning tree is minimized. Prove the correctness of your algorithm.