

Assignment 4
CSCI 3110: Design and Analysis of Algorithms
Due Jun 12, 2018

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Assignments are due on the due date before class and have to include this cover page. Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

This question studies another minimum spanning tree (MST) algorithm with running time $O(m \lg n)$ (matching the running time of Kruskal's algorithm). We assume that the input graph $G = (V, E)$ is connected to keep things simple. Since an MST $T = (V, E')$ is defined completely by its edge set E' (the vertex set is V), it suffices for the algorithm to output E' .

If G has only one vertex, then it has no edges. Thus, T does not have any edges either and we return $E' = \emptyset$.

If G has at least two vertices and thus at least one edge, then we compute E' as the union of two edge sets E_1 and E_2 . For every vertex $v \in G$, let e_v be the edge with minimum weight among the edges incident to v . Then $E_1 = \{e_v \mid v \in V\}$.

Next let $H = (V, E_1)$. Then construct a new graph G' that has one vertex v_C per connected component C of H . There exists an edge (v_{C_1}, v_{C_2}) in G' if and only if G contains an edge with one endpoint in C_1 and the other endpoint in C_2 . In this case, let e be the minimum-weight edge among all edges with one endpoint in C_1 and the other endpoint in C_2 . Then the edge (v_{C_1}, v_{C_2}) in G' has the same weight as e and stores a pointer $\text{orig}(v_{C_1}, v_{C_2}) = e$. To construct E_2 , we call the MST algorithm recursively to compute the edge set E'' of an MST of G' and, once this recursive call returns, set $E_2 = \{\text{orig}(e'') \mid e'' \in E''\}$.

Your task in this assignment is to prove that (1) this algorithm does indeed compute a minimum spanning tree and (2) its running is indeed $O(m \lg n)$.

Analysis.

- Argue that the edge set E_1 can be found in $O(m)$ time.
- Argue that the graph G' can be constructed in $O(m)$ time.
- Argue that the edge set E_2 can be constructed from the edge set E'' returned by the recursive call in $O(m)$ time.
- Argue that G' has at most half as many vertices as G .
- Argue that this implies that the algorithm takes $O(m \lg n)$ time.

Correctness. To simplify things, let us assume that no two edges of G have the same weight. (This assumption can be eliminated, but it's a bit technical.) Under this assumption, the MST T of G is unique. You do not need to prove this fact, even though you are welcome to convince yourself that this is indeed the case.

- Argue that every edge in E_1 belongs to T .
- Argue that every edge in E_2 belongs to T .
- Argue that $E_1 \cap E_2 = \emptyset$ and $|E_1| + |E_2| = n - 1$.
- Argue that this implies that $E_1 \cup E_2$ is exactly the edge set of T , that is, the algorithm is correct.

The first two claims can be shown using the Cut Theorem.