

Eksamensnoter - Amortized Analysis

André Oskar Andersen (wpr684)

March 26, 2020

23.1 - 1

Let (u, v) be a minimum-weight edge in a connected graph G . Show that (u, v) belongs to some minimum spanning tree of G

Make any cut that has (u, v) crossing it. Then, (u, v) must be a light edge, since it has minimum-weight, and thus also must be included in the MST

23.1 - 3

Show that if an edge (u, v) is contained in some minimum spanning tree, then it is a light edge crossing some cut of the graph

Let T_0 and T_1 be the two trees that are obtained by removing edge (u, v) from a MST. consider a cut which separates u from v . Suppose to a contradiction that there is some edge that has weight less than that of (u, v) in this cut. Then, we could construct a minimum spanning tree of the whole graph by adding that edge to $T_1 \cup T_0$. This would result in a minimum spanning tree that has weight less than the original minimum spanning tree that contained (u, v)

23.1 - 5

Let e be a maximum-weight edge on some cycle of connected graph $G = (V, E)$. Prove that there is a minimum spanning tree of $G' = (V, E - \{e\})$ that is also a minimum spanning tree of G . That is, there is a minimum spanning tree of G that does not include e .

Let A be any cut that causes some vertices in the cycle on one side of the cut, and some vertices in the cycle on the other. For any of these cuts, we know that the edge e is not a light edge for this cut. Since all the other cuts won't have the edge e crossing it, we won't have that the edge is light for any of those cuts either. This means that we have that e is not safe.

23.2 - 1

Kruskal's algorithm can return different spanning trees for the same input graph G , depending on how it breaks ties when the edges are sorted into order. Show that for each minimum spanning tree T of G , there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T .

Suppose that we wanted to pick T as our minimum spanning tree. Then, to obtain this tree with Kruskal's algorithm, we will order the edges first by their weight, but then will resolve ties in edge weights by picking an edge first if it is contained in the minimum spanning tree, and treating all the edges that aren't in T as being slightly larger, even though they have the same actual weight.

23.2 - 4

Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Kruskal's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W ? We can make Kruskal's algorithm run in $O(E\alpha(V))$, where α is inverse Ackermann function