# Eksamensnoter - Amortized Analysis

André Oskar Andersen (wpr684)

March 25, 2020

# 23 Minimum Spanning Trees

- We wish to find an acyclic subset $T \subset E$ that connects all of the vertices and whose total weight

$$w(T) = \sum_{(u,v) \in T} w(u,v)$$

  is minimized.

- Both Kruskal's algorithm and Prim's algorithm run in time $O(E \lg V)$ using ordinary binary heaps. By using Fibonacci heaps, Prim's algorithm runs in time $O(E + V \lg V)$, which improves over the binary-heap implementation if $|V|$ is much smaller than $|E|$

## 23.1 Growing a minimum spanning tree

- The two algorithms we onsider in this chapter use a greedy approach to the problem, although they differ in how they apply this approach. This greedy strategy is captured by the following generic method, which grows the minimum spanning tree one edge at a time. The generic method manages a set of edges $A$, maintaining the following loop invariant:

  > Prior to each iteration, $A$ is a subset of some minimum spanning tree

  At each step, we determine an edge $(u,v)$ that we can add to $A$ without violating this invariant, in the sense that $A \cup \{(u,v)\}$ is also a subset of a minimum spanning tree. We call such an edge a *safe edge* for $A$, since we can add it safely to $A$ while maintaining the invariant.

- A *cut* $(S, V - S)$ of an undirected graph $G = (V, E)$ is a partition of $V$. We say than an edge $(u,v) \in E$ *crosses* the cut $(S, V - S)$ if one of its endpoints is in $S$ and the other is in $V - S$.

- We say that a cut *respects* a set $A$ of edges if no edge in $A$ crosses the cut

- An edge is a *light edge* crossing a cut if its weight is the minimum of any edge crossing the cut. More generally, we say that an edge is a *light edge* satisfying a given property if its weight is the minimum of any edge satisfyinig the property.

- **Corollary 23.2**: Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function $w$ defiend on $E$. Let $A$ be a subset of $E$ that is included in some minimum spanning tree for $G$, and let $C = (V_C, E_C)$ be a connected component (tree) in the forest $G_A = (V, A)$. If $(u,v)$ is a light edge connecting $C$ to some other component in $G_A$, then $(u,v)$ is safe for $A$

## 23.2 The algorithms of Kruskal and Prim

**Kruskal's algorithm**

- Kruskal's algorithm finds a safe edge to add to the growing forest by finding, of all the edges that conenct any two trees in the fores, an edge $(u, v)$ of least weight.

**Prim's algorithm**

- Prim's algorithm has the property that the edges in the set $A$ always form a single tree. The tree starts from an arbitrary root vertex $r$ and grows until the tree spans all the vertices in $V$. Each step adds to the tree $A$ a light edge that connects $A$ to an isolated vertex - one on which no edge of $A$ is incident.