Minimum Spanning tree (Randomized Linear-time Algorithm)

Given $G=(V,E)$ and edge weights $w: E \to \mathbb{R}$, compute a spanning tree T, a subgraph of G (acyclic), with the minimum weight.

- When G is not connected, we say minimum forest of G.

- We know Kruskal's and Prim's algorithm running in $O(m\log n)$ and $O(m+n\log n)$ time, where $|V|=n$, $|E|=m$.

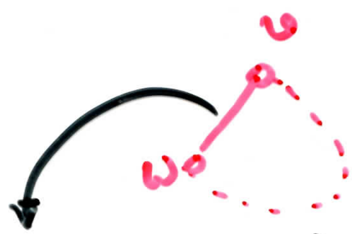
- We will examine another algorithm called Borůvka's algorithm which is suitable for use in a randomized algorithm.
- We aim for $O(m+n)$ expected running time.

Borůvka's algorithm

We use several simple facts about minimum spanning trees.

Fact 1. Let $v \in V$ be a vertex in G. MST of G must contain the edge (v, w) with the minimum weight among all edges incident to v.

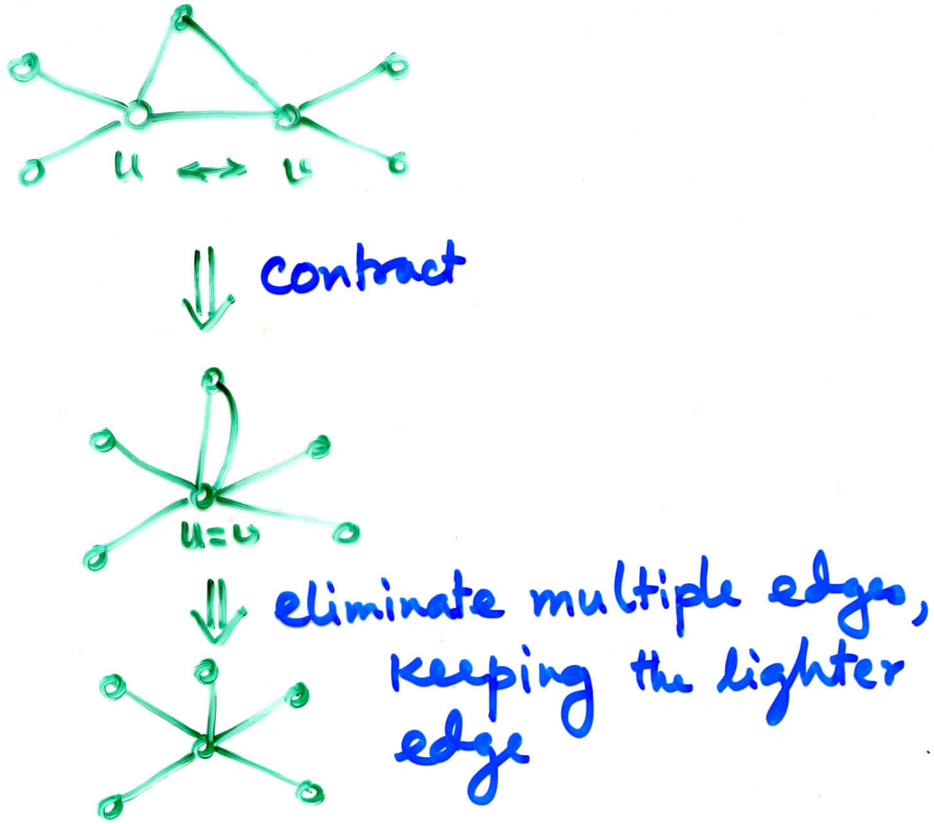
Proof. easy.



If not, adding vw makes lighter tree.

Edge Contractions.

A Borůvka-phase consists of contracting edges. Contracting an edge e means collapsing its endpoints into a single vertex and then eliminating any multiple edges.



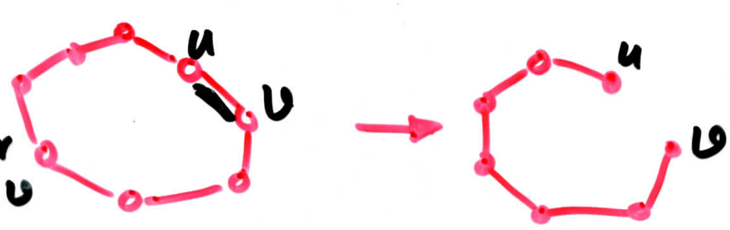
Contracting the minimum-weight edge incident to all vertices form one Borůvka-phase.

Fact 2. A Borůvka-phase can be implemented in $O(n+m)$ time.

Fact 3. The set of contracted edges in a Borůvka phase constitute a forest in G.

Proof.

if not, consider heaviest edge uv



uv cannot be a min-wt. edge either for u or v.

Fact 4. The contracted graph G' from G has at most $\frac{n}{2}$ vertices, and m edges.

Proof. Each contracted edge can be min. wt. edge of at most 2 vertices. Thus, the # of contracted edges is at least $\frac{n}{2}$ each reducing the vertex number by 1. #edges cannot be more than m since no new edge is created.

Borůvka's algorithm.

1. Identify the min. wt. edge for all vertices
2. Contract these edges to obtain graph G'
3. Compute a MST of G' recursively
4. $MST(G') \cup \{\text{contracted edges}\}$ constitute $MST(G)$.

The assertion in step 4 needs a proof.

- By fact 1, the contracted set of edges is in MST(G)

- Now consider contracting these edges in MST(G). We obtain a spanning tree of G' which has min. wt., namely MST(G').

Borůvka's algorithm runs in $O(m \log n)$ time because $T(m,n) = T(m, n/2) + \Theta(m+n)$

$$= \Theta(m \log n)$$

Randomization to speed up Borůvka's algorithm

Let F be a forest in G. ~~(spanning forest)~~
$w_F(u,v)$ be the maximum wt. of an edge on the path $u \rightsquigarrow v$ in F. It is ∞ if u, v are disconnected in F.

Def. An edge $(u,v) \in E$ is F-heavy if $w(u,v) > w_F(u,v)$. It is F-light otherwise.

Fact 5. An F-heavy edge cannot be in $MST(G)$.

Theorem Given a graph G and a forest F, all F-heavy edges in G can be computed in $O(n+m)$ time.

Random Sampling of MST.

The randomization comes in identifying and eliminating edges that are guaranteed not to be in the MST.

$G(p)$: the graph obtained from G by selecting each edge with probability $p > 0$. It has n vertices and mp expected number of edges.

- For reasonably large $p > 0$, we expect the minimum forest of $G(p)$ to be close to the MST of G
- This means there would be very few F-light edges in G in expectation.

- We need a little more concept from probability distribution.

- A random variable X has the negative binomial distribution with parameters n and p if it counts the number of trials required to have n success where each trial has a probability of success p.

- $E[X] = n/p$

- A random variable X stochastically dominates Y if for all $z \in \mathbb{R}$, $\Pr[X > z] \geq \Pr[Y > z]$. $\Rightarrow E[X] \geq E[Y]$.

- If X and Y ~~are~~ have negative binomial distribution with parameters n_1, p and n_2, p where $n_1 \geq n_2$, then X stochastically dominates Y.

Fact 6. Let F be a MSF in $G(p)$. Then, the number of F-light edges in G is stochastically dominated by a random variable X having a negative binomial distribution with parameters n and p.

$\Rightarrow E[\#\, F\text{-light edges in } G] \leq \frac{n}{p}$.

Proof. See the book (page 300) of Motwani–Raghavan on randomized algorithms.

- One can think of F-light edges being chosen for MSF of $G(p)$ with probability p.
- At most $(n-1)$ edges need to be selected or $(n-1)$ success needs to be made ~~when~~ each trial has prob. p.
- This is stochastically dominated by n success trials with prob. p.

Linear-time Algorithm.

Step 1. - Apply Boruvka-phase 3 times interleaved with edge contractions.
- The resulting graph G_1 has at most $n/8$ vertices. Let C be the set contracted edges
- If G_1 is empty, return C

Step 2. $G_2 = G_1(p)$ is computed where $p = \frac{1}{2}$

Step 3. Recursively compute minimum Spanning forest F_2 of G_2.

Step 4. Using linear-time verification, identify F_2-heavy edges in G_1 and delete them. to obtain G_3.

Step 5. Recursively compute the minimum Spanning forest of G_3. Let it be F_3.

Step 6. Output $C \cup F_3$.

Time analysis.

Let $T(n, m)$ be the expected time for $G=(V,E)$ with $n=|V|$, $m=|E|$.

- Step 1 runs in $O(n+m)$ time and produces G_1 with $\leq \frac{n}{8}$ vertices and m edges
- Step 2 produces $G_2 = G_1(1/2)$ with $n/8$ vertices and $m/2$ edges (in expectation). It also takes $O(n+m)$ time.
- Step 3 takes $T(n/8, m/2)$ time in expectation
- Step 4 verifies in $O(n+m)$ time and produces G_3 with at most $\frac{n}{8}$ vertices and an expected number of edges at most $\left(\frac{n}{8}\right) \| (1/2) = \frac{n}{4}$ (Fact 6).
- Step 5 takes $T(n/8, \frac{n}{4})$ expected cost
- Step 6 takes $O(n)$ time to output.

- $T(n,m) \leq T(n/8, m/2) + T(n/8, n/4) + c(n+m)$.

By induction one can show

$T(n,m) \leq 2c(n+m)$ completing the proof.