— Boruvka's MST algorithm —

Below, we assume the input graph $G(V, E)$ is a connected positive integer edge-weighted undirected simple graph. We also assume edge weights of the input graph are pairwise distinct. Given G, we want to find the MST of G.

- Observation 1: For any vertex $v \in V$ of G, the MST of G must contain the edge that has the minimum weight among all the edges incident to v.

  Proof: Due to the MST cut property.

- Observation 2: Marking the minimum weighted edge incident to v at each vertex v of G independently does not cause marked edges to form any cycle. Hence, vertices of G with all such marked edges is a spanning forest of G.

  Proof: Suppose there is a cycle C in the graph induced by marked edges. Consider the maximum weighted edge $(u, v) \in C$. W.l.o.g., suppose (u, v) was marked by v. But the weight of marked edge (v, v') on C is strictly less than $w(u, v)$, contradicting the least weighted edge incident to v is (u, v).

- Definition: The contraction of a set S of nodes of an edge weighted undirected simple graph $G(V, E)$ is an edge weighted undirected graph G/S wherein

  $V - S \cup \{v\}$ is the vertex set of G/S for a supernode v such that $v \notin V$, and

  for every edge $e(v', v'') \in G$, an edge $e'(w', w'')$ is introduced into G/S, where $w' = v$ (resp. $w'' = v$) if $v' \in S$ (resp. $v'' \in S$) otherwise $w' = v'$ (resp. $w'' = v''$).

  Note that contracting vertex set S of G could cause a bunch of self-loops and multiedges, each of which is incident to v, in G/S.

- Observation 3: Corresponding to every edge (u', v') of G/S, there is a unique edge $(u, v) \in G$ such that $u \in u', v \in v'$, and $u, v \in V$.

- Pseudocode:

  (a) For every node v of G, make a greedy choice by marking the least weighted edge e incident to v, and output e.

  (b) For each connected component CC of the graph induced by marked edges in G,

  (i) $G \leftarrow G/S$, where S is the set of nodes of CC.

  (c) Compute simple graph G' from G: remove self-loops in G; for every two nodes v', v'' of G, if there are more than one edge between v' and v'', then remove all the edges between v' and v'' except for the one with least weight.

  (d) If the number of nodes in G' is greater than one, rename G' as G, and go to Step (a).

  Below, a phase denotes the first three steps of this procedure.

- Proof of correctness:

- Since the number of nodes at least halves from any phase to the phase subsequant to it, the procedure does terminate.

- Just after Step (a), graph induced by marked edges is spanning forest of G. That is, just before contracting (super)nodes belonging to any connected component in Step (b), there is a tree spanning those nodes.

- Just before termination, i.e., once all phases are executed, since Step (d) ensures G' has exactly one supernode, the graph induced by marked edges in G is a spanning tree of G.

- For any supernode v, consider the cut $C(A, V - A)$, where A is the set of nodes of G that result from recursively uncontracting v. Since v is marking the light-edge of C, from the MST cut property, this marked edge is guaranteed to belong to the MST of G. Hence, G' is a minimum spanning tree of G.

- The Boruvka's algorithm takes $O(m \lg n)$ time in the worst-case:

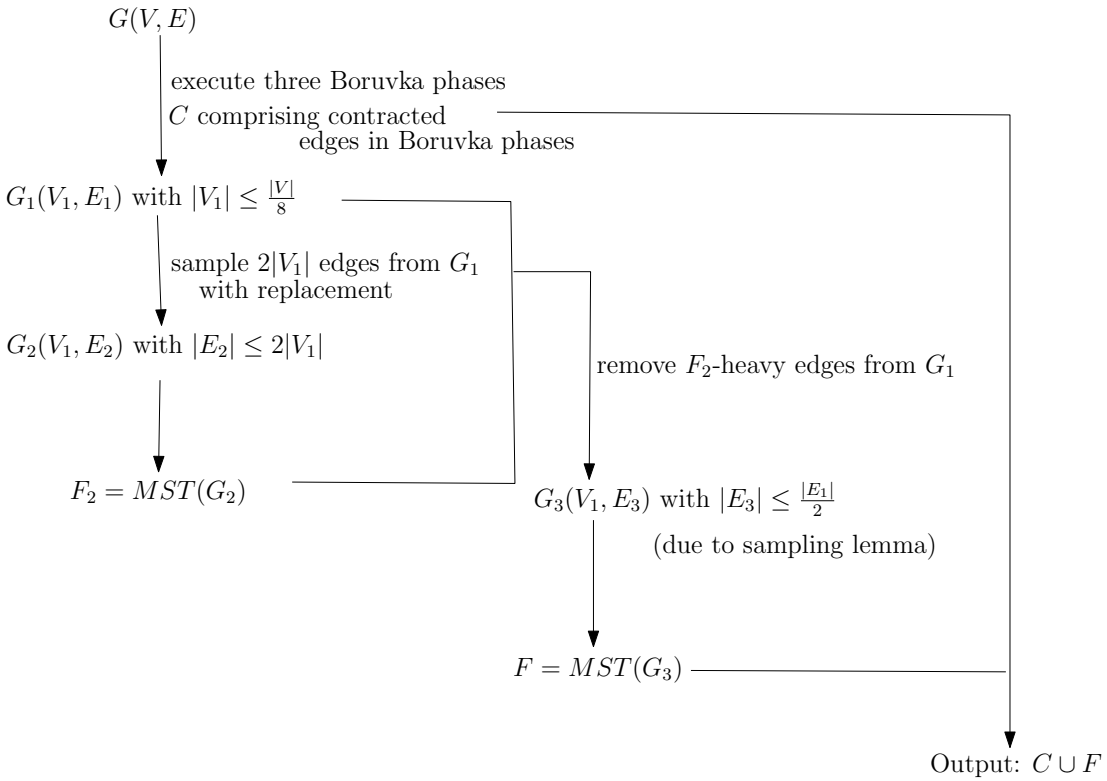
- If a phase i starts with n' vertices and e' edges, phase $i + 1$ starts with at most $\frac{n'}{2}$ vertices and at most e' edges. Hence, the number of phases is $O(\lg n)$.

- Further, each phase takes $O(m)$ time: identifying connected components w.r.t. marked edges takes $O(m+n)$ time; forming an adjacency list corresponding to simple graph takes $O(m + n)$ time; since G is connected, $m \geq n - 1$.

- Though the algorithm takes $O(m \lg n)$ time in the worst-case, typically, number of nodes drop more than a factor of two from one phase to the next phase. Due to this reason, in practice, Boruvka's algorithm runs faster. Many ideas in this algorithm are used in MST algorithms invented later, ex. in $O(m \lg \lg n)$ time Yao's algorithm, $O(m \lg^* n)$ time Fredman-Tarjan's algorithm, and in Karger's expected linear time algorithm. Besides, Boruvka's algorithm is parallelizable.

— KKT's[1] expected linear time MST algorithm —

- Let subgraph F of G be a forest. Then, an edge $e(u, v) \in G$ is *F-heavy* if w_e is larger than the maximum weighted edge on the unique path from u to v in F. Otherwise, e is *F-light*. Note that e is F-light even if u-v path does not exist in F. The F-heavy and F-light edges are useful in exploiting the MST cycle property.
- Below figure describes this algorithm. Since each Boruvka phase reduces the number of vertices by at least half, three Boruvka phases are applied to input grpah $G(V, E)$ so that the number of vertices in G_1 is at most $\frac{|V|}{8}$. The edges in E_1 are randomly sampled so that $|E_2|$ is at most $2|V_1| = \frac{|V|}{4}$. Since the number of vertices and the number of edges in G are reduced, MST of G_2 can be computed efficiently. Indeed, on G_2, with a recursive call, MST F_2 of G_2 is computed. The F_2 helps in pruning F_2-heavy edges from G_1, and leading to a set E_3 comprising F_2-light edges in G_1. The sampling lemma, stated below, helps in upper bounding $|E_3|$, which turns out to be $\frac{|E_1|}{2}$. With yet another recursive invocation, MST of G_3 is computed. Since the edges removed in sampling are F_2-heavy, and since G_3 retains all the F_2-light edges in G_1, the edges in $MST(G_3)$ together with the marked edges in Boruvka phases together are guaranteed to yield a MST of G.



MST-KKT(G(V, E)):

(i) if $|V|$ and $|E|$ are of $O(1)$, compute a MSF F' by brute-force; return F'

[1] named after its inventors Karger, Klein, and Tarjan

(ii) $G_1(V_1, E_1) \leftarrow$ apply three Boruvka phases to G

also, let C be the set of contracted edges in these phases

(iii) $G_2(V_1, E_2) \leftarrow$ sample a set E_2 of $2|V_1|$ edges from G_1, independently, uniformly at random, and with replacement

(iv) $F_2 \leftarrow$ MST-KKT(G_2)

(v) $G_3(V_1, E_3) \leftarrow$ using a linear-time MST verification algorithm, delete F_2-heavy edges in G_1 from G_1

(vi) $F \leftarrow$ MST-KKT(G_3)

(vii) return $C \cup F$

The step (v) is accomplished by using a linear-time algorithm for MST verification: given a graph G_1 and a spanning forest F_2, the MST verification algorithm by Komlos' determines the set E_3 of F_2-light edges in G_1. Essentially, KKT's algorithm uses ideas from Boruvka's, utilizes the services of a MST verification algorithm, and exploits random sampling.

This procedure is guaranteed to terminate since the number of vertices are reducing in each recursive call. We know from Boruvka's algorithm, each contracted edge in a Boruvka phase is guaranteed to belong to $MST(G)$. Due to MST cycle property, none of the deleted F_2 heavy-edges are part of $MST(G)$. Hence, it is easy to see the output graph is spanning, acyclic, and connected.

- Observation: Let E_2 be a set of randomly sampled edges of a graph $G_1(V_1, E_1)$. Any edge $e \in E_1$ is F_2-light iff $e \in MST(E_2 \cup \{e\})$.

  Sampling Lemma: For a random sample $E_2 \subseteq E_1$ of edges of a graph $G_1(V_1, E_1)$, the expected number of F_2-light edges is at most $\frac{|V_1||E_1|}{|E_2|}$.

  Proof: It suffices to show $pr[e \text{ is } F_2\text{-light}] \leq \frac{|V_1|}{|E_2|}$, for any randomly sampled edge e from E_1. From the backward analysis,

  $$\begin{aligned} & pr[e \text{ is } F_2\text{-light} \mid \text{fixed } E_2 \cup \{e\}] \\ &= pr(e \in MST(E_2 \cup \{e\}) \mid \text{fixed } E_2 \cup \{e\}) \quad \text{(due to the above observation)} \\ &\leq \tfrac{|V_1|-1}{|E_2 \cup \{e\}|} \\ &\leq \tfrac{|V_1|}{|E_2|}. \end{aligned}$$

  Hence, $pr[e \text{ is } F_2\text{-light}]$

  $$\begin{aligned} & = pr[e \text{ is } F_2\text{-light} \mid \text{fixed } E_2 \cup \{e\}] \cdot pr[\text{fixed } E_2 \cup \{e\}] \\ & \leq pr[e \text{ is } F_2\text{-light} \mid \text{fixed } E_2 \cup \{e\}] \\ & \leq \tfrac{|V_1|}{|E_2|}. \end{aligned}$$

- By choosing $|E_2| = 2|V_1| \leq 2(\frac{n}{8}) = \frac{n}{4}$, from the sampling lemma, the expected number of F_2-light edges in G_1 is at most $\frac{|E_1|}{2} \leq \frac{m}{2}$. The latter is due to $|E_1| \leq |E|$. Hence, $|E_3|$ is at most $\frac{m}{2}$.

Since each Boruvka phase halves the number of vertices, the three Boruvka phases together lead to $|V_1| \leq \frac{|V|}{8} = \frac{n}{8}$. Further, the number of vertices in G_1, G_2, and G_3 are equal.

- Let $T(n, m)$ be the expected running time of MST-KKT on graph G with n vertices and m edges. Since step (i) takes $O(m+n)$ time, since step (iii) takes $O(m+n)$ time, and since the MST verification algorithm takes $O(m+n)$ time, the recurrence (ignoring base cases) is, $T(n, m) \leq T(\frac{n}{8}, \frac{n}{4}) + T(\frac{n}{8}, \frac{m}{2}) + O(m+n)$. The first term on the right side of this recurrence is for the recursive invocation on G_2 and the second term is due to the recursive call on G_3. Solving this recurrence with the guess and substitute method, the expected running time is $O(m + n)$. The tighter analysis shows that this algorithm indeed takes $O(m)$ expected time.

References:

R. Motwani and P. Raghavan, Randomized Algorithms. Cambridge University Press, 1995.