Karger–Klein–Tarjan vs. Kruskal for Minimum Spanning Trees

Implementation and Benchmark Report

1 Problem and Model

Given a connected, undirected, weighted graph $G = (V, E, w)$ with $|V| = n$, $|E| = m$, the task is to output a minimum spanning tree (MST). We work in the comparison-based RAM model with unit-cost edge-weight comparisons.

2 Algorithms (structured description)

Karger–Klein–Tarjan (KKT), expected $O(m + n)$.

1. **Base case.** If $n \leq n_0$ or $m \leq m_0$, return Kruskal's MST.
2. **Two Borůvka rounds.** For each current supervertex, pick its lightest incident edge, add it to the partial MST, and contract all chosen edges (delete self-loops; keep only the lightest among parallel edges). After two rounds, the vertex count drops by a constant factor. Let C be the set of chosen edges and $G_1 = (V_1, E_1)$ the contracted graph.
3. **Sampling.** Independently include each edge of E_1 with probability $p = \frac{1}{2}$ to obtain subgraph H.
4. **Sample MSF.** Compute an MSF (Minimum Spanning Forest) F of H (we use Borůvka inside the recursion).
5. **F-heavy filtering.** An edge $e = (u, v)$ is F-heavy if $w(e)$ exceeds the maximum edge weight on the unique F-path between u and v (or $+\infty$ if they are disconnected). Using a linear-time MST verifier (Borůvka tree plus $O(n)$ Farach–Colton–Bender LCA with parent/max ladders), mark all F-heavy edges and delete them to get $G_2 = (V_1, E_2)$ of F-light edges.
6. **Recurse and combine.** Recursively compute an MSF T_{G_2} of G_2. Output $C \cup T_{G_2}$, expanding contractions to original vertices.

Kruskal + union–find serves as the deterministic baseline and correctness oracle for KKT.

Kruskal + union–find, $O(m \log m)$. Sort edges by weight; scan and add an edge iff it connects different DSU components. Complexity is dominated by sorting.

3 Foundational Properties

Lemma 1 (Cut property)**.** *For any nonempty proper $S \subset V$, every minimum-weight edge crossing the cut $(S, V \setminus S)$ belongs to every MST of G.*

Proof. Fix such an S and let e be a minimum-weight cut edge. Any spanning tree T must contain some edge f crossing this cut. Consider replacing f by e in T. Connectivity is preserved because both cross $(S, V \setminus S)$. If $w(e) < w(f)$, the replacement yields a strictly lighter spanning tree, contradicting the minimality of any MST omitting e. If $w(e) = w(f)$, the replacement preserves minimal weight but still shows e can appear in some MST. Therefore every MST contains e. □

Lemma 2 (Borůvka edges)**.** *Every edge chosen in a Borůvka round lies in every MST of the current graph; contracting them preserves the set of MSTs.*

Proof. Each chosen edge is the minimum across a singleton cut $\big(\{v\}, V \setminus \{v\}\big)$, so Lemma 1 forces its inclusion in every MST. Contracting such mandatory edges merges endpoints already guaranteed to be connected in all MSTs, so MST structure on the contracted graph matches that of the original. □

Lemma 3 (Cycle property via F-heaviness)**.** *Let F be an MSF of a subgraph. If u and v are connected in F and an edge $e = (u, v)$ satisfies $w(e) >$ max edge weight on the F-path between u and v, then e is absent from every MST of the current graph.*

Proof. Adding e to F creates a unique cycle. By construction, e is the heaviest edge on that cycle. The cycle property of MSTs states that the heaviest edge on any cycle cannot appear in an MST, so e is excluded from all MSTs. □

4 Correctness of KKT

Theorem 1. *KKT returns an MST of G.*

Proof. We prove the claim by induction on $n + m$.

Base case. If $n \leq n_0$ or $m \leq m_0$, the algorithm runs Kruskal, which is correct by standard arguments.

Inductive step. Assume correctness for smaller $n + m$. Run two Borůvka rounds: by Lemma 2, the set C of selected edges must be in every MST, and contracting C reduces the instance to $G_1 = (V_1, E_1)$ without altering the MST family.

Sampling forms H and its MSF F. Apply Lemma 3: delete every F-heavy edge of G_1 to obtain $G_2 = (V_1, E_2)$. Because only edges forbidden by the cycle property are removed, the MST set of G_2 equals that of G_1.

Invoke the inductive hypothesis on G_2: the recursive call returns an MSF T_{G_2}, which is therefore an MST of G_1. Finally, expand contractions and union T_{G_2} with C. Since C is mandatory and T_{G_2} spans the contracted graph optimally, their union is an MST of the original graph G. □

5 Expected Linear Time

Lemma 4 (Sampling lemma)**.** *Sampling each edge independently with probability p and letting F be an MSF of the sample yields $\mathbb{E}[\#F\text{-light edges in } G] \leq \frac{n}{p}$.*

Proof. Order edges of G by nondecreasing weight and simulate Kruskal on the sample. For edge e, let A_e be the event "e is F-light when processed," and let X_e indicate "e enters F." If A_e holds,

$\Pr[X_e = 1 \mid A_e] = p$ (it is sampled and joins distinct components); otherwise $\Pr[X_e = 1 \mid A_e^c] = 0$. Hence $\mathbb{E}[X_e] = p \Pr[A_e]$, and summing gives

$$p \sum_{e \in E} \Pr[A_e] \;=\; \sum_{e \in E} \mathbb{E}[X_e] \;=\; \mathbb{E}[|F|] \;\leq\; n - 1.$$

Therefore $\sum_e \Pr[A_e] \leq n/p$, which equals $\mathbb{E}[\#F\text{-light edges}]$. □

Theorem 2. *KKT runs in expected $O(m + n)$ time.*

Proof. Charge work to edges at each recursion node.

- **Sampled children.** Each edge is included with probability $1/2$, so expected total sampled edges across all levels is $\sum_{i \geq 0} (1/2)^i m \leq 2m$.
- **Filtered children.** The filtered graph contains only F-light edges. Lemma 4 with $p = \frac{1}{2}$ gives $\mathbb{E}[|E_2|] \leq 2|V_1|$ at a node with $|V_1|$ vertices. Two Borůvka rounds shrink $|V_1|$ by a factor of at least 4, so the expected sum of filtered edges over all depths is a convergent geometric series $O(n)$.

Summing these disjoint edge streams, the expected total work is $\Theta(m + n)$. □

6 Implementation and Benchmarks

- Binary: `mst_bench` from `mst/src/main_new.cpp` (`g++ -std=c++17 -O2`).
- Config: $n \in \{2^{10}, \ldots, 2^{17}\}$; $m \in \{2n, 4n, 8n, 16n\}$; 3 graphs $\times$ 3 reps; deterministic seeds.
- Verification: Borůvka tree plus $O(n)$ Farach–Colton–Bender LCA and parent/max ladders to classify F-heavy edges.

7 Figures and Discussion

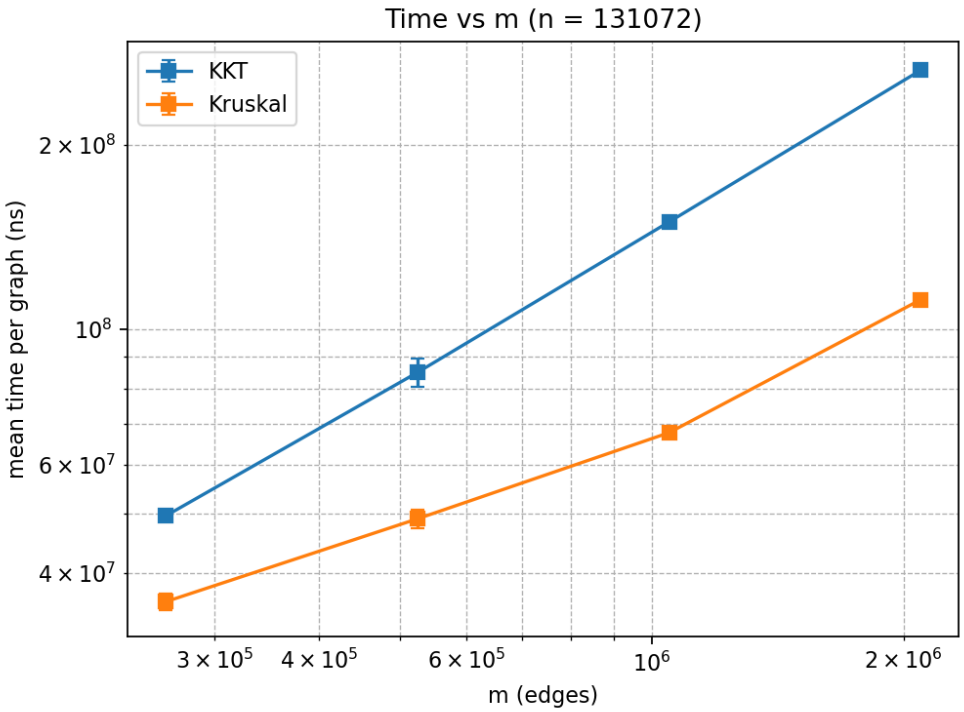

Figure 1: Fixed $n = 131072$: time per graph vs. m.

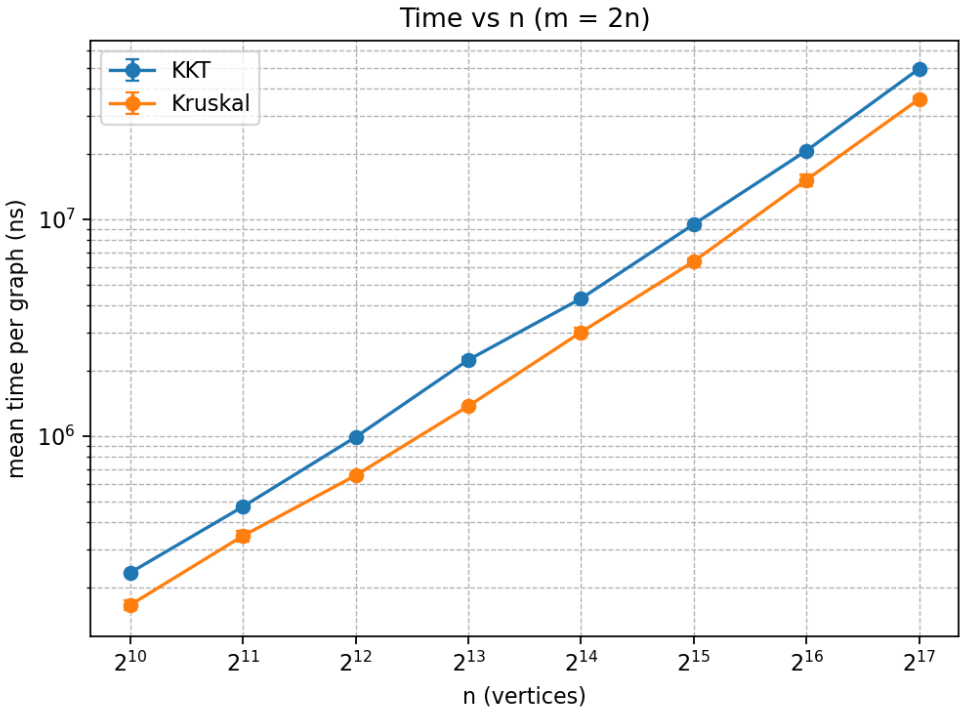

Figure 2: Varying n with density $m = 2n$.

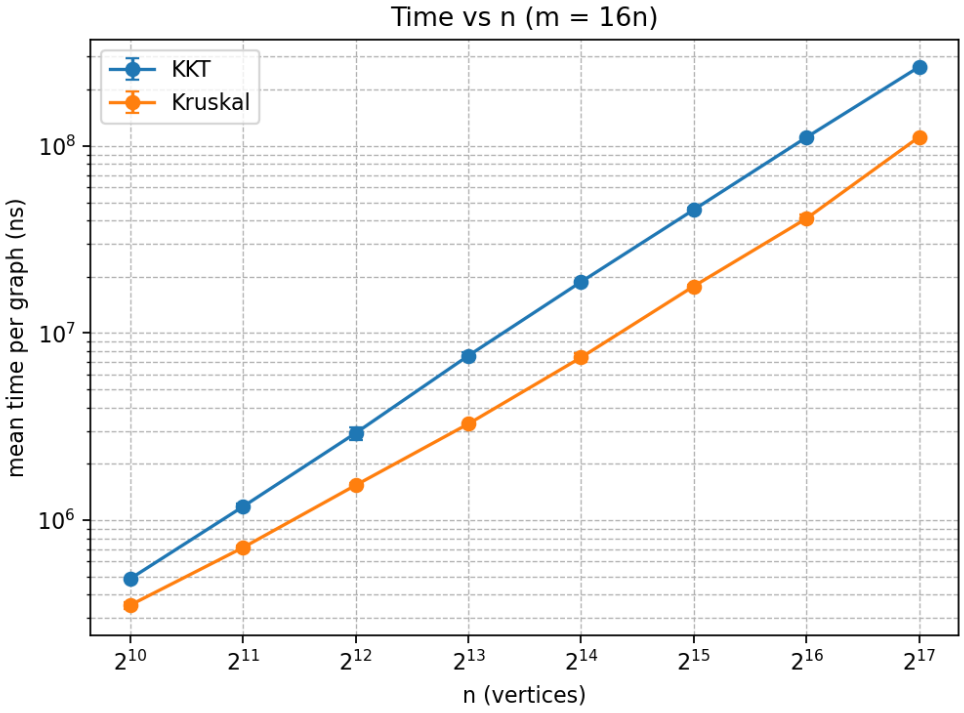

Figure 3: Varying n with density $m = 16n$.

Observations.

- **Scaling with m (Fig. 1).** Kruskal grows roughly with m (sort-dominated); doubling m roughly doubles time. KKT is consistently slower and its gap widens as density rises.
- **Scaling with n at $m = 2n$ (Fig. 2).** Kruskal's curve is smooth with modest growth; KKT mirrors the shape but remains far slower, reflecting high constant factors despite linear expectation.
- **Scaling with n at $m = 16n$ (Fig. 3).** Higher density amplifies the gap: KKT's verification/contraction overheads dominate, while Kruskal continues to benefit from cache-friendly sort + DSU.
- **Practical takeaway.** The theoretically near-linear KKT is "galactic" in practice because of heavy verification constants and memory traffic; optimized $O(m \log n)$ Kruskal remains the pragmatic choice.

8 Applications of MST

- **Network design.** Cost-minimal communication, power, and transport backbones.
- **Clustering.** Single-linkage hierarchies and density-based methods use MSTs to expose cluster structure.
- **Vision.** Graph-based image segmentation (e.g., Felzenszwalb–Huttenlocher) partitions via MST cuts.
- **VLSI/CAD.** Clock-tree and interconnect synthesis start from MST backbones before Steiner refinements.
- **Computational biology.** Parsimonious phylogenetic backbones and scaffolding heuristics on distance graphs.