Data Structures and Algorithms

Lecture 27

Aniket Basu Roy

2026-04-01 Wed

Contents

1 Agenda

1.1 Correctness of MST Algorithms

- Cut Property: statement and proof

- Correctness of Prim-Jarník via Cut Property
- Correctness of Kruskal's via Cut Property
- Handling duplicate edge weights

2 The Cut Property

2.1 Theorem (Cut Property)

Let $G = (V, E, w)$ be a connected graph with **distinct** edge weights. Let $(S,\ V \setminus S)$ be any cut, and let

$$e^* = \operatorname{argmin} \{\, w(e) \mid e \in E(S,\ V \setminus S) \,\}$$

be the minimum-weight edge crossing the cut. Then e^* belongs to **every** MST of G.

3 Proof of the Cut Property

Let T be any MST of G. We show $e^* \in T$.

Suppose for contradiction that $e^* \notin T$. Write $e^* = (u, v)$ with $u \in S$, $v \in V \setminus S$.

Since T is a spanning tree, there is a **unique path** P from u to v in T.

P must cross the cut $(S, V \setminus S)$ at least once. Let $f \in P$ be an edge of T that crosses the cut.

Since e^* is the **minimum** cut-set edge and f also crosses the cut, and all weights are distinct:

$$w(e^*) < w(f).$$

Form $T' = (T \cup \{e^*\}) \setminus \{f\}$.

- **Connected:** removing f splits T into two components; adding e^* reconnects them (both cross the same cut).
- **Acyclic:** adding e^* to T creates a unique cycle through P; removing $f \in P$ breaks it.
- Hence T' is a spanning tree with $w(T') = w(T) - w(f) + w(e^*) < w(T)$.

This contradicts T being an MST. ■

4 Correctness of Prim-Jarník

4.1 Claim

Prim-Jarník correctly computes an MST.

4.2 Proof

At each step, let S be the set of vertices already extracted from Q.

The algorithm picks

$$e^* = \operatorname{argmin} \{ w(u,v) \mid u \in S,\, v \notin S \},$$

the minimum-weight edge in the cut-set of $(S,\, V \setminus S)$.

By the **Cut Property**, e^* is in every MST.

Induction: every edge added is an MST edge. After $|V| - 1$ additions we have a spanning tree, hence the (unique) MST. ■

5 Correctness of Kruskal's Algorithm

5.1 Claim

Kruskal's algorithm correctly computes an MST.

5.2 Proof

Consider the step where Kruskal's algorithm adds edge $e = (u, v)$. At this point u and v are in different components; let C_u be the component of u.

Define the cut $(C_u,\ V \setminus C_u)$.

e **is the minimum-weight edge crossing this cut:** Any lighter edge crossing this cut would have been processed earlier. Such an edge e' was either

- added to A (so its endpoints merged into the same component — impossible if it truly crosses $(C_u, V \setminus C_u)$), or
- skipped (formed a cycle, meaning both endpoints were already in the same component, so it does not cross the cut).

Contradiction.

By the **Cut Property**, $e \in$ every MST. All $|V| - 1$ edges added by Kruskal's are MST edges, so the result is the MST. ■

6 Handling Duplicate Edge Weights

The Cut Property as stated requires distinct weights.

6.1 Perturbation Argument

Replace each $w(e)$ with $w(e) + \varepsilon_e$ for sufficiently small, distinct $\varepsilon_e > 0$.

- For small enough perturbations, any MST of the perturbed graph is also an MST of the original.
- The perturbed weights are distinct, so the Cut Property applies.
- Kruskal's and Prim-Jarník remain correct; break ties by any consistent rule.

6.2 Remark

A more careful version of the Cut Property holds without the distinctness assumption: the minimum cut edge is in **some** MST (not necessarily every). Both algorithms still return a valid MST.

7 Questions

- Is the MST unique when all edge weights are distinct? (Yes — the strict inequality in the proof forces uniqueness.)
- Can there be multiple MSTs? (Yes, when some edge weights coincide.)
- What are the time complexities of Kruskal's and Prim-Jarník? (Next two lectures.)