Data Structures and Algorithms

Lecture 26

Aniket Basu Roy

2026-03-30 Mon

Contents

1 Agenda

1.1 Minimum Spanning Trees

- Edge-weighted graphs
- Definition of MST, Example
- Cuts and cut-sets
- Kruskal's algorithm
- Prim-Jarník algorithm

2 Edge-Weighted Graphs

- A graph $G = (V, E, w)$ where $w : E \to \mathbb{R}$ assigns a real weight to each edge.
- The **weight** of a set of edges $F \subseteq E$: $w(F) = \sum_{e \in F} w(e)$.

3 Spanning Trees

Given a connected undirected graph $G = (V, E)$:

- A **spanning tree** is a subgraph $T = (V, E')$ that is a tree (connected and acyclic).
- Any spanning tree has exactly $|V| - 1$ edges.
- Removing any edge from a spanning tree disconnects it; adding any non-tree edge creates exactly one cycle.

4 Minimum Spanning Tree

Given a connected edge-weighted graph $G = (V, E, w)$, a **minimum spanning tree** is a spanning tree T^* that minimizes total weight:

$$T^* = \underset{T \text{ spanning tree}}{\arg\min} \; w(T).$$

4.1 Example

```
Vertices: {a, b, c, d, e}
Edges:  (a,b):3  (a,c):1  (b,c):4  (b,d):2  (c,d):5  (c,e):6  (d,e):7
MST edges: (a,c):1,  (b,d):2,  (a,b):3,  (c,e):6   [total = 12]
```

5 Cuts and Cut-Sets

5.1 Cut

A **cut** of $G = (V, E)$ is a partition $(S,\ V \setminus S)$ where $\emptyset \subsetneq S \subsetneq V$.

5.2 Cut-Set

The **cut-set** (crossing edges) of cut $(S, V \setminus S)$:

$$E(S,\, V \setminus S) \;=\; \{\, (u,v) \in E \;\mid\; u \in S,\ v \in V \setminus S \,\}.$$

5.3 Key Fact

Every spanning tree T contains at least one edge from every cut-set. (Removing all cut-set edges disconnects the graph, so any spanning subgraph must keep at least one.)

6 Kruskal's Algorithm

6.1 Idea

Greedily add the cheapest edge that does not form a cycle.

6.2 Pseudocode

```
KRUSKAL(G, w)
1.  A = {}
2.  for each vertex v in G.V
3.      MAKE-SET(v)
4.  sort edges in E by weight (non-decreasing)
5.  for each edge (u, v) in E  (in sorted order)
6.      if FIND-SET(u) != FIND-SET(v)
7.          A.insert((u, v))
8.          UNION(u, v)
9.  return A
```

6.3 Trace on Example

Process edges in order: (a,c):1, (b,d):2, (a,b):3, (b,c):4, (c,d):5, (c,e):6, (d,e):7

- Add (a,c):1 — no cycle.
- Add (b,d):2 — no cycle.
- Add (a,b):3 — no cycle.
- Skip (b,c):4 — would form cycle a-b-c-a.
- Skip (c,d):5 — would form cycle a-b-d-c-a.
- Add (c,e):6 — no cycle. Now $|A| = 4 = |V| - 1$. Done.

7 Prim-Jarník Algorithm

7.1 Idea

Grow a tree from an arbitrary root by always adding the cheapest edge leaving the current tree.

7.2 Pseudocode

```
PRIM(G, w, r)
1.  for each u in G.V
2.      u.key = infinity
3.      u.parent  = NIL
4.  r.key = 0
5.  Q = G.V               // min-priority queue keyed by u.key
6.  while Q != {}
7.      u = EXTRACT-MIN(Q)
8.      for each v in G.Adj[u]
9.          if v in Q and w(u,v) < v.key
10.             v.parent   = u
11.             v.key = w(u,v)   // DECREASE-KEY in Q
```

At termination, the MST edges are $\{(v, v.\pi) \mid v \neq r\}$. Note $v.\pi$ and `v.parent` are used interchangeably.

7.3 Invariant

Let S = vertices already extracted from Q. The edges $\{(v, v.\pi) \mid v \in S \setminus \{r\}\}$ form an MST of $G[S]$. v.key = minimum weight of any edge from v to a vertex in S (or ∞ if none).

8 Questions

- How do we prove correctness of Kruskal's and Prim-Jarník? (Next lecture: Cut Property.)
- What data structures are used for MAKE-SET / FIND-SET / UNION? (Lect 29: Union-Find.)