MST: Minimum Spanning Tree

Given \( G = (V, E) \), with weights

so let a min \( G \) tree, \( T = (V, E') \), \( E' \subseteq E \)

\[
\text{cost}(T) = \sum_{e \in E'} w(e)
\]

Prim's Algorithm

1) sort edges \( G \) in increasing order of weight

\[ |E| \log |E| \]

2) Add edges in sorted order, \( y \) exactly one end point is in \( T \)

\[
\sum \text{cost} 3
\]
Prim's:

Invariant:
The current partial MST is always connected.

Pick a start vertex \( u \)

- \( C_M(u_0) = 0 \)
- \( \gamma(u) \) all other vertices = \( \infty \)

0(\( |V| \)) Insert all vertices into a PQ (Priority Queue)

While PQ not empty

- \( u = \text{deleqmin}(PQ) \)
- For all neighbors \( x \) of \( u \)
  \( \gamma(x) > \gamma(x) \)

\[
\begin{array}{c|c|c}
A & 0 & 0 \\
\hline
\infty & -\infty & \times \\
B & 8 & y \\
\end{array}
\]
\[ y (x) > w(u, x) \]
\[ c(u, x) = w(u, x) \]
\[ \text{decrease key } (x, c(u, x)) \]

\[ O(|V|) \text{ delete min} \quad | \quad \text{binary tree PQ} \]
\[ O(|E|) \text{ decrease key} \]
\[ O(|V| \log |V|) \text{ cry} \]
\[ O((|V| + |E|) \log |V|) \]

---

Why does greedy strategy work for MST?

CUT property:
If we have a cut, then we can always choose the smallest edge that crosses the cut.

\[ G : \text{ graph} \]
\[ S \subset V : \text{ subset of vertices} \]
\[ (S, V - S) \text{ is a cut (or partition) of } G \]
Prove: Given \( G_j \) and given a partial MST, \( T \).

1. Select a cut \( T \) with respect to \( T \).
2. Pick the smaller weight edge \( e \) and add to \( T \).
3. We will show that \( T' \) is part of some MST.

\[ \text{Prove: Given } G_j \text{ and given a partial MST, } T \]

\[ T' = T \cup \{ e \} \]

We will show that \( T' \) is part of some MST.

\[ \omega(e) \leq \omega(e) \]
Case 1: the is only one edge that cross the ar e1 in the only one, it is the smallest as well e1 has to be selected.

Case 2: if there are more edge crossing the ar show how selecting the least weight edge is OK!

Consider $T'$:

$$\text{Cost}(T') = \text{Cost}(T_1) + \text{Cost}(T_2) + \omega(e_2)$$

Find $T'$ that includes $e_1$.

$$T' = T_1 \cup \{e_1\} \cup T_2$$

$$\text{Cost}(T') = \text{Cost}(T_1) + \text{Cost}(T_2) + \omega(e_1)$$

$$\text{Cost}(T') \le \text{Cost}(T)$$

Because $\omega(e_1) \le \omega(e_2)$

but $T$ is an MST, so at least $\text{Cost}(T)$

$$\text{Cost}(T') = \text{Cost}(T)$$
Kruskal’s Algorithm

we maintain a “forest”

Pick set of edge in increasing order of weights

Current “tree” can be disconnected, but there are no cycles

\[ \text{SORT} \]

\[ \checkmark \text{ac} = 1 \]
\[ \checkmark \text{ef} = 1 \]
\[ \checkmark \text{cd} = 2 \]
\[ \checkmark \text{ab} = 4 \]

\[ a - 3x \]
a) sum all edges

b) keep on adding edges in sorted order \( O(|E|) \)

1) check for cycle, each time! \( \text{DFS on the forest} \)

2) stop when we have a single tree & all vertices have been added.

Sorhig \( O(|E| \log |E|) \)

\[ |E| = O(|V|^2) \]

\[ \log |E| = \log |V|^2 = (2 \log |V|) \]

\[ O(|E|) \leftarrow \text{outer for loop} \]

\[ \text{detect cycle for each edge} \]

\[ \text{DFS: } O(|V| + |E|) \]

\[ = O(|V| + |V|) \]

\[ = O(|V|) \]

Cost:

\[ |E| \log |V| + |E| \cdot |V| \]

Sorhig + Cycle

\[ \text{Pimb} \]

Final MST has exactly \( |V| - 1 \) edge, \( V \) vertex.

\( |E'| = |V| - 1 \) edge, \( V \) vertex.

\[ \frac{\text{Pimb}}{|V| \log |V| + |E| \log |V|} \]
Correctness: Our property is given a cut that repeats the connector, adding the lower edge that crosses the cut, always leads to some MST.

Union-Find data structure:

Graph:

- a - c
- e - f
- c - d

Find operation: Are they part of the same component? $O(\log |V|)$ time.

Union operation: $O(1)$ merge 2 components into 1.

$|E| \log |V| + O(|E| \log |V|)$
\[ |E| \log |V| + O(|E| \log |V|) \]

\[ \sum_{e \in E} \quad \text{for each edge} \quad \text{find operations on cycles,} \]

\[ = O(|E| \log |V|) \quad \text{Kruskal's time.} \]