Midterm 2
Mon 7pm - 9pm
Covers everything after mid 1 to last wk cheat sheet

Min Spanning Tree (MST)

Given weighted undirected graph \(G = (V,E),\)
\(w: E \rightarrow \mathbb{R}_+\),
find a connected subgraph that includes all vertices minimizing total wt

![Diagram of a graph with nodes and weights](image)

one sol’n: \(5 + 9 + 7 + 28 = 49\)
better sol’n: \(5 + 7 + 2 + 21 = 35\) optimal

Appl. network design

Observe: The opt subgraph must be acyclic.

PDF Sketch: [Diagram of an undirected, acyclic, connected graph]
idea 0 - brute force \implies exponential
idea - greedy!

Kruskal's Alg'nm (1956): High-level version

1. \( T = \emptyset \)
2. repeat {
   pick next smallest-weight edge \( e \)
3. if \( T \cup \{ e \} \) doesn't contain a cycle
4. insert \( e \) to \( T \)
}

\textbf{Implementation: (detailed vers.)}

1. sort edges in increasing order of weight
2. create set \( \{ u \} \), \( u \in V \)

\( O(m \log n) \) time
2. Create set \( \{ u \} \), \( u \notin V \)
3. For each edge \( uv \) in sorted order
4. If \( u \) & \( v \) are in different sets
5. Output \( uv \) & union the two sets

**Snapshot of T:**

**Lines 4-5:** "union-find" data structure
- Union two disjoint sets: \( O(1) \) time
- Find set containing \( v \): \( O(\alpha(n)) \) time (amortized)
  \( \alpha(n) \ll \log \log \log \ldots \log n \)

\[ \Rightarrow O(m\alpha(n)) \text{ time} \]

**Total time** \[ O(m \log n) \]

**Correctness Pf:** (Assume wts are all distinct)

**Key Lemma**
Given any subset \( S \subseteq V \),
smallest-weight edge \( e \) between \( S \) & \( V-S \)
must be in the MST, \( T^* \).
Pf: By contradiction.
Suppose \( e \notin T^* \).
\( T^* \cup \{e\} \) contains a cycle \( C \).

\( C \) must contain another edge \( e' \) between \( S \) & \( V-S \).

\[
T^* \cup \{e\} - \{e'\} \text{ is a tree with weight } w(T^*) + w(e) - w(e') \leq w(T^*)
\]

because \( w(e) < w(e') \):
Contradiction! \( \square \)

Correctness Pf for Kruskal:
- each edge \( e = uv \) inserted to \( T \)
  is in MST \( T^* \) by Key Lemma.

each edge \( e = uv \) \( u \in S \) is in MST \( T^* \) by Key Lemma.

### Prim's Alg'm (1957): High-level vers.

\[
S = \{s\}, \quad T = \emptyset
\]
while \( S \neq V \) do <

pick edge \( uv \) with \( \min \frac{w(uv)}{w(uv)} \) \( u \in S, \ v \in V-S \)

insert \( uv \) to \( T \)
insert \( v \) to \( S \)

}\)

**Correctness Pf:** by Key Lemma again!

\[
S = a
\]
\[
\Rightarrow \quad a \quad b
\]
like Dijkstra, can be implemented using Fibonacci heaps in $O(n \log n + m)$ time

Other Algs:

Boruvka (1926) $O(m \log n)$

Yao '75 $O(m \log \log n)$

Fredman, Tarjan '85 $O(m \log^* n)$

Grabow et al. '86 $O(m \log(\log^* n))$

Karger, Klein, Tarjan '94 $O(m)$ randomized

Chatelle '97 $O(m \alpha(n))$

OPEN $O(m)$ det.?