# Minimum Spanning Tree

Reading: CLS Chapter 23; Erikson Chapter 7

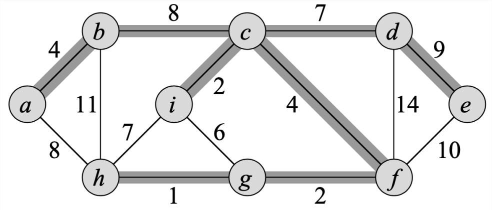
Jiaming Jiang

ECS 122A Algorithm Design and Analysis – Winter 2021

# Minimum Spanning Tree (MST)

- Undirected connected weight graph $G = (V, E, w)$
- **Spanning tree $T \subseteq E$:** a tree $T$ that connects all vertices; "spans" the graph
- **Minimum spanning tree (MST) $T \subseteq E$:** $w(T) = \sum_{(u,v) \in T} w(u, v)$ is minimized

Example: $w(T) = 37$



# Distinct Edge Weight(s)

- MST is not necessarily unique.
- Special case: Consider a graph in which every edge has weight 1. Then every spanning tree is a MST with weigth $|V| - 1$.
- If all edge weights in a connected graph $G$ are distinct, then $G$ has a unique MST.
- For simplicity in theory, assume all edge weights distinct, and therefore, has a unique MST.

# Generic MST Algortihm

- Greedy algorithm: grows the MST $A$ one edge at a time – safe edge
- Maintain an invariant: $A$ is a subset of some MST
- **Safe edge**: an edge of minimum-weight that can be added to $A$ without violating the invariant

GENERIC-MST$(G, w)$

$A = \emptyset$<br>
**while** $A$ does not form a spanning tree<br>
    find an edge $(u, v)$ that is safe for $A$<br>
    $A = A \cup \{(u, v)\}$<br>
**return** $A$

# Proving the Two Properties of a Greedy Algorithm

1. Suboptimality: A MST contains MSTs.

Let $T$ be a MST of $G = (V, E, w)$. Removing $(u, v)$ of $T$ partitions $T$ into two trees $T_1$ and $T_2$. Then $T_1$ is a MST of $G_1 = (V_1, E_1)$ and $T_2$ is a MST of $G_2 = (V_2, E_2)$.

*Sketch of Proof:*

$$w(T) = w(T_1) + w(u, v) + w(T_2)$$

There cannot be a better subtree thatn $T_1$ or $T_2$, otherwise $T$ would not be optimal.

# Proving the Two Properties of a Greedy Algorithm

2. Greedy choice

Let $T$ be a MST of $G = (V, E, w)$, $A \subseteq T$ a subtree of $T$, and $(u, v)$ a minimum-weight edge in $G$ connecting the subgraphs induced by $A$ and $T - A$. Then $(u, v) \in T$.

*Sketch of Proof:*

If $(u, v) \notin T$, then

- $(u, v) \cup T$ forms a cycle;
- replacing one of the edges of $T$ by $(u, v)$ forms a new tree $T'$;
- this is contradiction to $T$ being a MST.

# Prim's Algorithm

- Starts from an arbitrary root vertex $r$ and grows until the tree $A$ spans all the vertices
- $v.key$: min weight of any edge connecting $v$ to a vertex in $A$
- $v.\pi$: the parent of $v$ in $A$
- Data structure: priority queue

# Prim's Algorithm

```
MST-PRIM(G, w, r)
for each u ∈ G.V
    u.key = ∞
    u.π = NIL
r.key = 0
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
        if v ∈ Q and w(u, v) < v.key
            v.π = u
            v.key = w(u, v)
```

# Prim's Algorithm: Example