Shortest Path

- We need to prove that greedy choice works.
- That is repeatedly select one of the vertex $u \in V - S$ with the minimum shortest path estimate to be included in S.
- Relax all edges leaving u when it is included in S.
- We need to prove that

  $d[v]$ is the value of the shortest path from s to v when the latter is inserted into S.

Shortest Path

- Proof by induction on size of S.
- The induction hypotheis is

  For each $v \in S$, $d[v] = dist[v]$, and for all $v \in V - S$, $d[v]$ is equal to the length of the shortest path from s to v that wholly lie within S except for v itself.

- Basis is $|S| = 1$. The shortest path from s to itself is 0. Initially, only $d[s] = 0$, and a path from s to other vertices $v \in V - S$ that wholly lie within S would consist of the vertices connected by direct edges to s. So $d[v]$ is properly set by initialization code.

Shortest Path

- Induction step: consider S when v is being inserted.
- Assume that $d[v] > dist(s, v)$ when v is inserted into S.
- Let $y \in V - S$ be the first vertex on the actual shortest path $s \longrightarrow v$.
- By optimal substructure property, $dist(s, v) = dist(s, y) + dist(y, v)$
- This means, $dist(s, y) < d[v]$, and the shortest path to y lies wholly within S except y itself.
- Before v is included, by inductive hypothesis $d[y] < d[v]$.
- So, greedy choice must select y before v.
- This is a contradiction.

Kruskal's MST

- Optimal substructure property exhibited by Kruskal's MST
- Consider a generic tree T. Let T be a MST of a given graph G.
- Delete an edge (u, v) from T, let the two trees created be T_1 and T_2.
- Claim:
  - T_1 is a MST of subgraph induced by vertex set V_{T_1}
  - T_2 is a MST of subgraph induced by vertex set V_{T_2}

Kruskal's MST

- We have

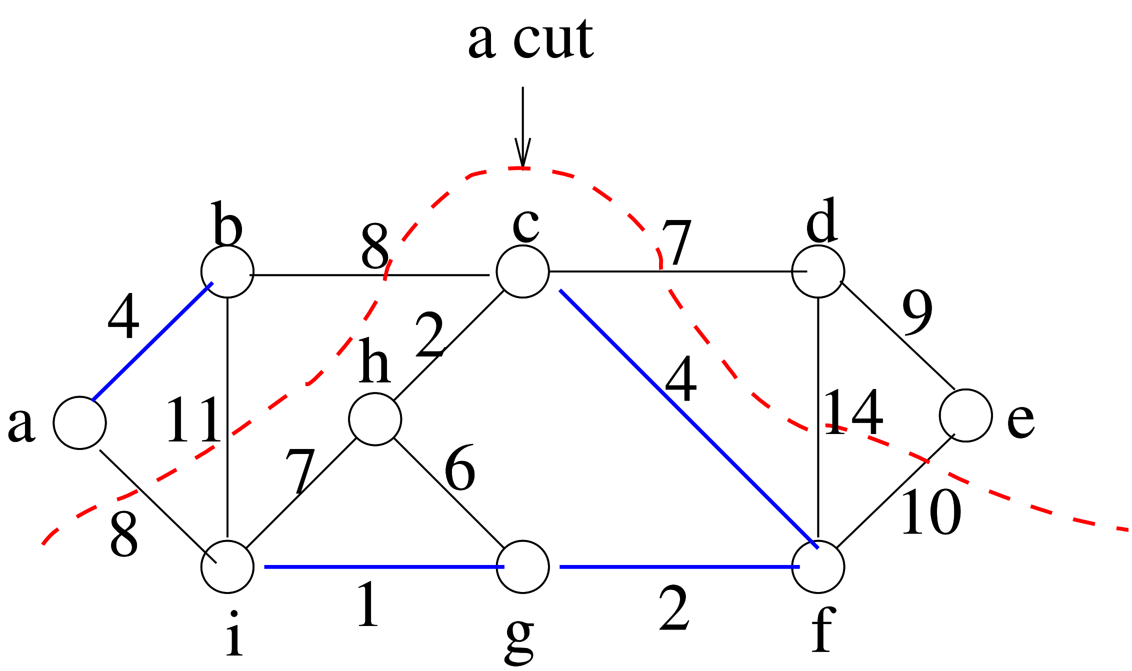
$$w(T) = w(T_1) + w(T_2) + w(u, v)$$

- So if T_1 is not a MST of (V_{T_1}, E_{T_1}) or T_2 is not a MST of (V_{T_2}, E_{T_2}), then we can replace these trees by MST and still join these new pair of trees using (u, v).
- the new tree formed as such definitely has lower weight than T.
- So T is not a MST.

MST: Greedy Choice Property

- Let A be subset of edges in an MST.
- A cut is a set of edges that partitions the vertices into two subsets.
- A cut C respects A provided no edge of A belongs to C.

MST: Greedy Choice Property



no edge of cut is a blue edge

cut respects set of blue edges

S={a, b, d, e}

V–S ={c, f, g, h, i}

MST: Greedy Choice Property

- Let T be the MST that includs the set of edges A.
- Consider C a cut respecting A.
- Let (u, v) be the `light edge` (with minimum cost) joining a vertex of $V - S$ with that of S.
- Greedy choice always includes `light edge`.
- Let T excludes (u, v).
- We show that there is another MST T' that includes $A \cup \{(u, v)\}$

MST: Greedy Choice Property

- Since T excludes (u, v) – inserting it will create a cycle in T.
- Therefore, another edge (x, y) of cut C is included in T.
- Now form

$$T' = (T - \{(x, y)\}) \cup \{(u, v)\}$$

- Thus T' includes $A \cup \{(u, v)\}$.
- Also we know (u, v) is a `light` edge of the cut.
- So, $w(u, v) \leq w(x, y)$.
- Hence,

$$w(T') = w(T) - w(x, y) + w(u, v) \leq w(T).$$

implying T' must also be an MST.