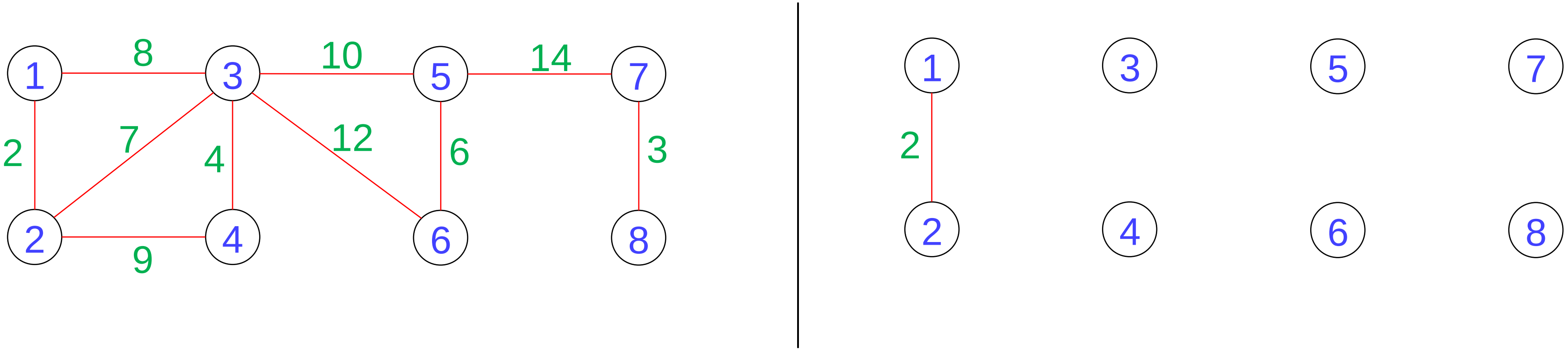
Kruskal's Algorithm

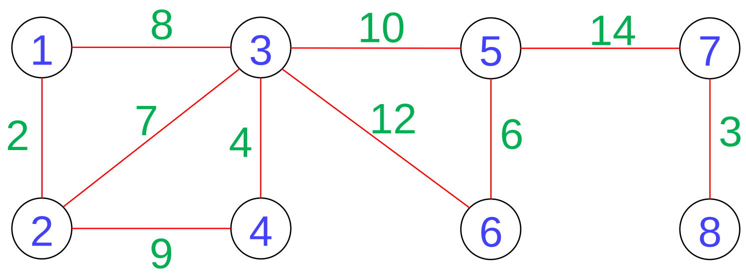
Kruskal’s Algorithm

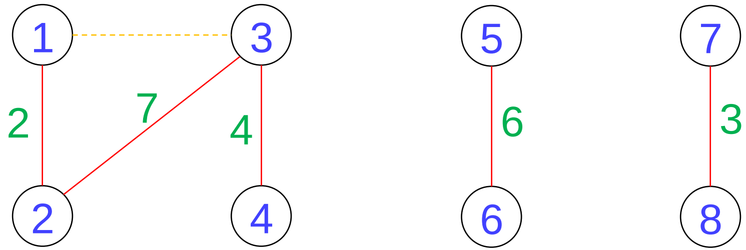
- Start with a forest that has no edges
- Consider edges in ascending order of cost
- Edge (1, 2) is considered first and added to the forest



Kruskal’s Algorithm

- Edge (7, 8) is considered next and added to the forest
- Edge (3, 4) is considered next and added to the forest
- Edge (5, 6) is considered next and added to the forest
- Edge (2, 3) is considered next and added to the forest
- Edge (1, 3) is considered next and rejected because it creates a cycle

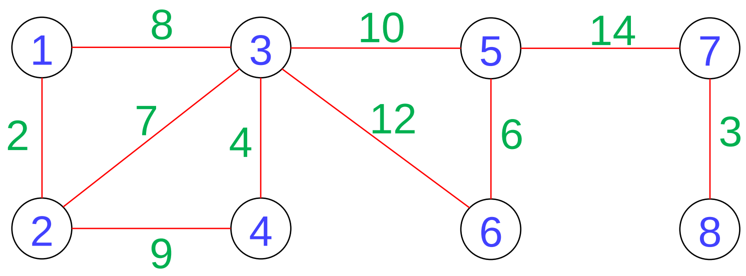


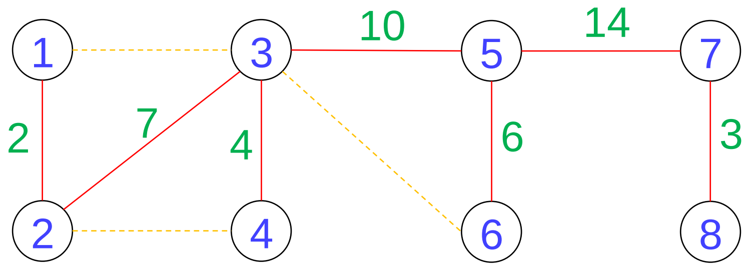


Kruskal’s Algorithm

- Edge (2, 4) is considered next and rejected because it creates a cycle
- Edge (3, 5) is considered next and added to the forest
- Edge (3, 6) is considered next and rejected because it creates a cycle
- Edge (5, 7) is considered next and added to the forest

- ***N - 1*** edges have been selected and no cycle formed, so we must have a spanning tree



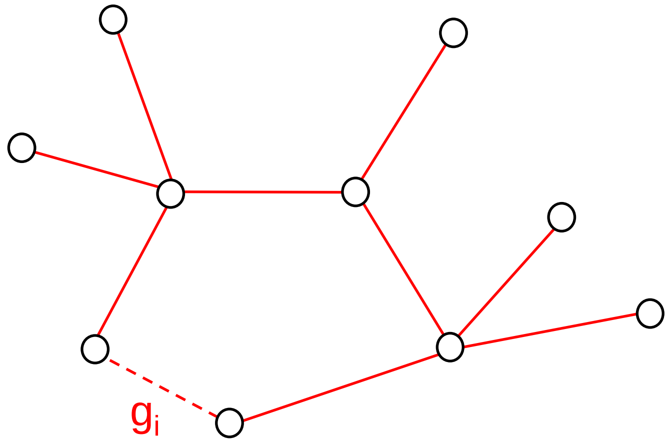


- Cost is 46, minimum cost spanning tree is unique when all edge costs are different

Correctness Analysis of Kruskal's Algorithm

- **Assumption:** All edge lengths are distinct
- Let's Kruskal's algorithm has picked the set of edges: $\mathbf{g_1 < g_2 < g_3 < \ldots\ldots < g_i < \ldots\ldots g_{V-1}}$
- Let the optimum/best tree be: ....$\mathbf{f_1 < f_2 < f_3 < \ldots\ldots < f_i < \ldots\ldots f_{V-1}}$

- Let this set differ and the first place they differ is: ***i***
- Two cases are: $\mathbf{g_i < f_i}$ and $\mathbf{g_i > f_i}$
- <u>First case: $\mathbf{g_i < f_i}$</u>
- Then $\mathbf{g_i} \notin \{\mathbf{f_1 < f_2 < f_3 < \ldots\ldots < f_{i-1}}\}$
- So, $\mathbf{g_i}$ doesn't belong to the optimum tree
- Lets add $\mathbf{g_i}$ in the optimum tree, which forms a cycle ***C***
- Can $\mathbf{g_i}$ be the longest edge in the cycle? No!
- $\mathbf{g_i}$ can't be that edge whose length is greater than the sum of the length of all other edges in the cycle, because these edge would form a cycle in our Kruskal's algorithm too

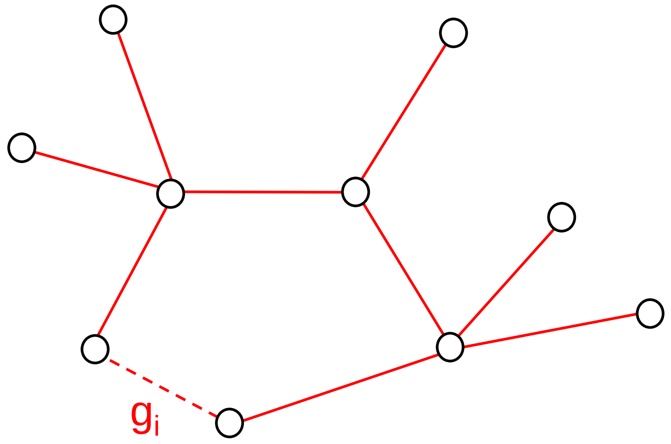


Contradiction !!!

Correctness Analysis of Kruskal's Algorithm

- **Assumption:** All edge lengths are distinct
- Let's Kruskal's algorithm has picked the set of edges: $\mathbf{g_1} < \mathbf{g_2} < \mathbf{g_3} < \ldots\ldots < \mathbf{g_i} < \ldots\ldots\ \mathbf{g_{N-1}}$
- Let the optimum/best tree be: ....$\mathbf{f_1} < \mathbf{f_2} < \mathbf{f_3} < \ldots\ldots < \mathbf{f_i} < \ldots\ldots\ \mathbf{f_{N-1}}$

- Let this set differ and the first place they differ is: ***i***
- Two cases are: $\mathbf{g_i} < \mathbf{f_i}$ and $\mathbf{g_i} > \mathbf{f_i}$
- <u>Second case: $\mathbf{f_i} < \mathbf{g_i}$</u>
- Then $\mathbf{f_i} \notin \{\mathbf{g_1} < \mathbf{g_2} < \mathbf{g_3} < \ldots\ldots < \mathbf{g_{i-1}}\}$
- Why did Kruskal's algorithm didn't pick $\mathbf{f_i}$
  - Because $\mathbf{f_i} \cup \{\mathbf{g_1} < \mathbf{g_2} < \mathbf{g_3} < \ldots\ldots < \mathbf{g_{i-1}}\}$ contains a cycle
  - Then $\mathbf{f_i} \cup \{\mathbf{f_1} < \mathbf{f_2} < \mathbf{f_3} < \ldots\ldots < \mathbf{f_{i-1}}\}$ contains a cycle
- Then optimum tree also has a cycle



Contradiction !!!

Kruskal's Algorithm

- **Pseudocode**
- **The Important Step**
- How to check if including an edge ***(u, v)*** forms a cycle or not?
  - Iff ***u*** and ***v*** are in the same connected ***components***
  - For a forest containing ***k*** number of edges, the number of trees is ***V – k***
  - Kruskal's algorithm pick an edge at a time and include in the tree
  - In every step, the number of connected components decreases by 1 by including an edge

Sort edges in increasing order of cost of edges

$e_1, e_2, e_3, e_4, \ldots e_E$ such that $c(e_i) \leq c(e_{i+1})$

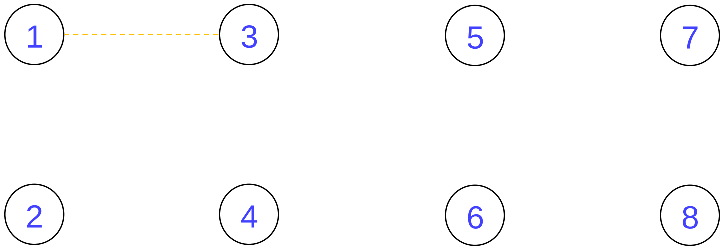
T = Φ

for i = 1 to E **do**

  if ($\{e_i\} \cup T$) is a tree

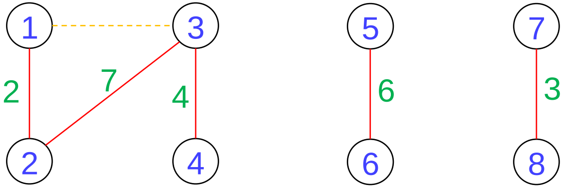
    T = $\{e_i\} \cup T$

return T

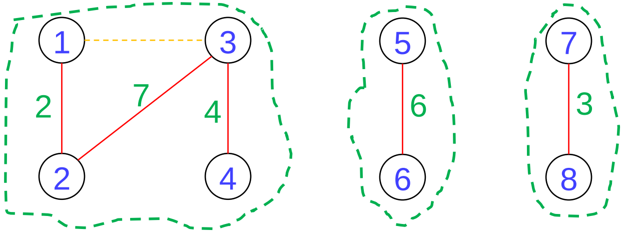


Data Structures for Kruskal's Algorithm

- Does the addition of an edge (***u***, ***v***) to ***T*** result in a cycle?



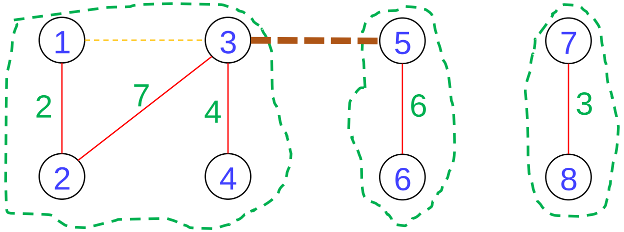
- Each component of ***T*** is a tree
- When ***u*** and ***v*** are in the same component, the addition of the edge (***u***, ***v***) creates a cycle
- When ***u*** and ***v*** are in the different components, the addition of the edge (***u***, ***v***) does not create a cycle



- Each component of ***T*** is defined by the vertices in the component
- Represent each component as a set of vertices

{1, 2, 3, 4}, {5, 6}, {7, 8}

- Two vertices are in the same component iff they are in the same set of vertices



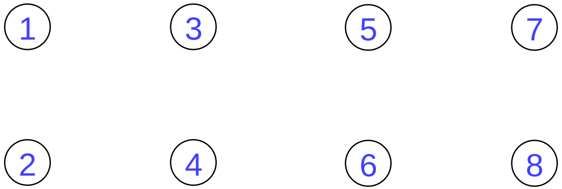
- When an edge (***u***, ***v***) is added to ***T***, the two components that have vertices ***u*** and ***v*** combine to become a single component

Data Structures for Kruskal's Algorithm

- In our set representation of components, the set that has vertex ***u*** and the set that has vertex ***v*** are united

{1, 2, 3, 4} + {5, 6} → {1, 2, 3, 4, 5, 6}

- Initially, ***T*** is empty



- Initial sets are:

{1} {2} {3} {4} {5} {6} {7} {8}

- Does the addition of an edge (***u***, ***v***) to ***T*** result in a cycle? If not, add edge to ***T***

s1 = find(u); s2 = find(v);

if (s1 != s2) union(s1, s2);

Next Lecture

Kruskal's Algorithm

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in, shrutilipi.bhattacharjee@tum.de