CS 304
Design and Analysis of Algorithm

Minimum Spanning Tree
(Kruskal's Algorithm)

Last Class's Topic

- Minimum Spanning Tree (Prim's Algorithm)
  - What is spanning tree & Minimum spanning tree?
  - What will happen if we add an edge in a minimum spanning tree?
- Read the following topics from "Cormen-pg563"
  - Cut
  - Cross
  - Light edge

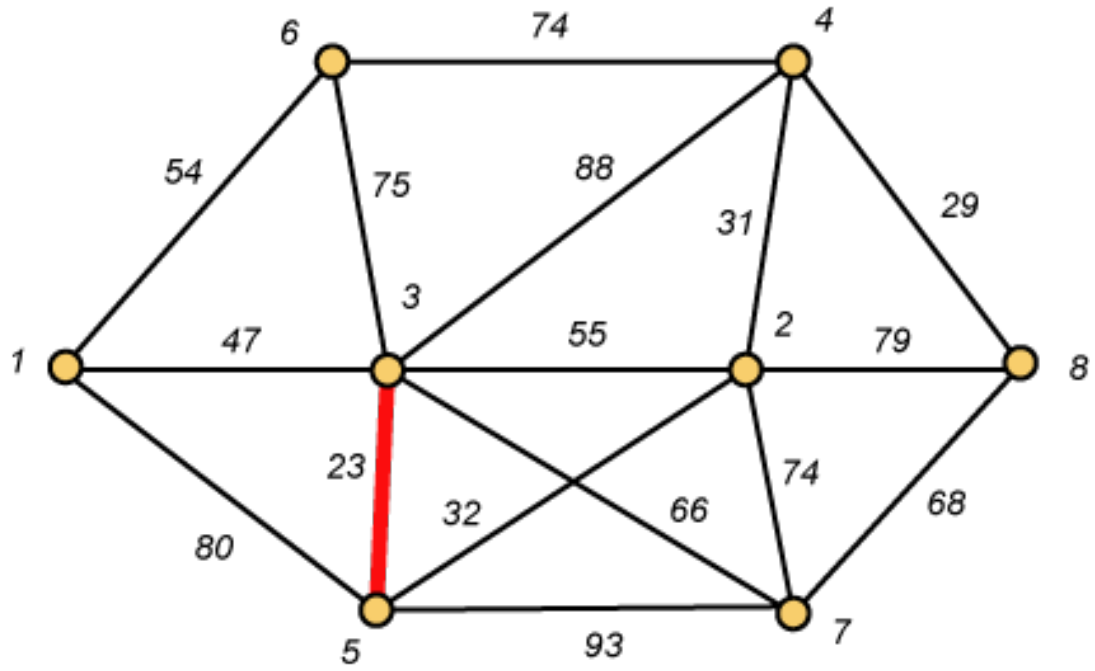
Some Definition

- Cut:
  - Partition of V. Ex: (S, V-S)
- Cross:
  - Edge (u,v) crosses the cut (S, V-S) if one of its endpoints is in S and the other is in V-S.
- Light edge:
  - An edge crossing a cut if its weight is the minimum of any edge crossing the cut.

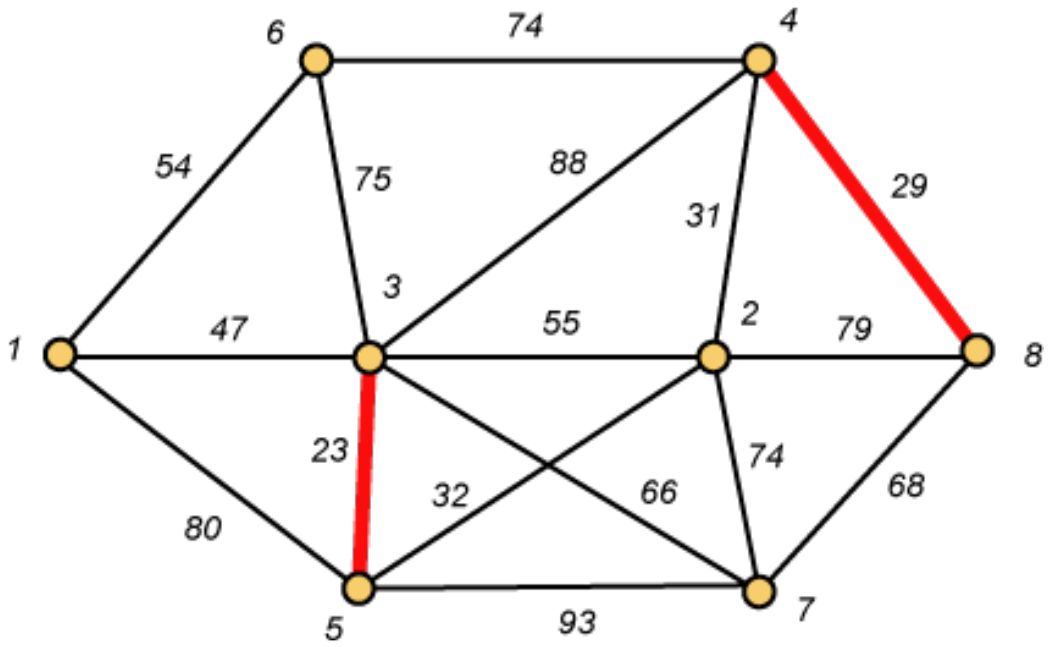
Kruskal's Algorithm

- Edge based algorithm
- Add the edges one at a time, in increasing weight order
- The algorithm maintains *A* – a **forest of trees**. An edge is accepted it if connects vertices of distinct trees
- We need a data structure that maintains a partition, i.e.,a collection of disjoint sets
  - MakeSet(S,x)
  - Union(S_i,S_j)
  - FindSet(S, x)

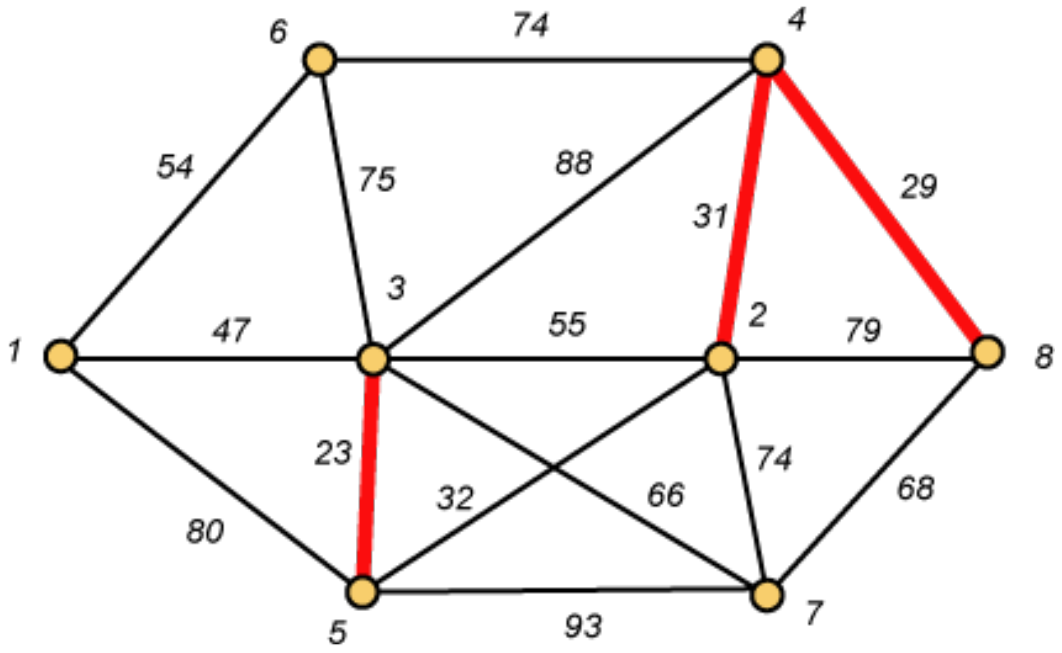
Kruskal – Step 1



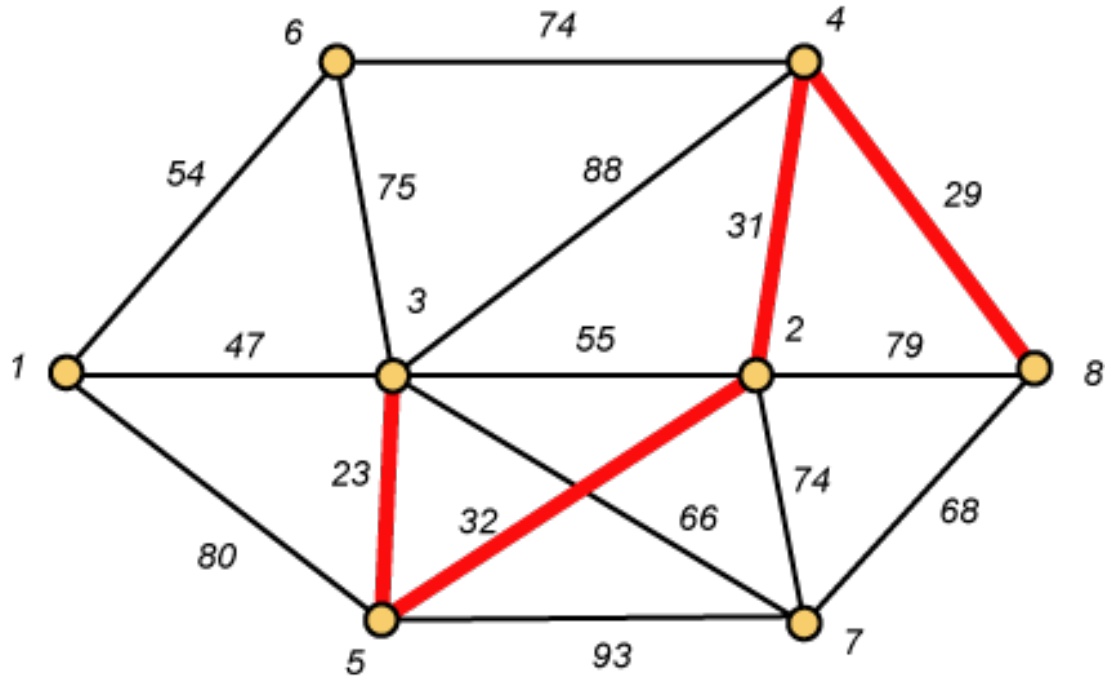
Kruskal – Step 2



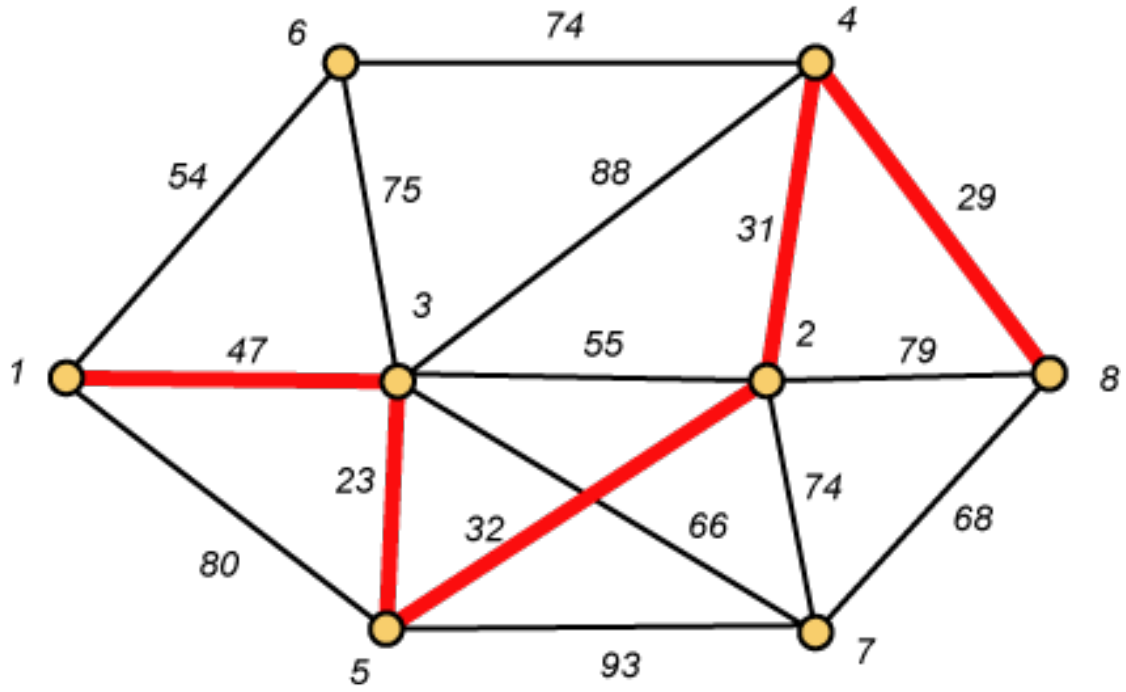
Kruskal – Step 3



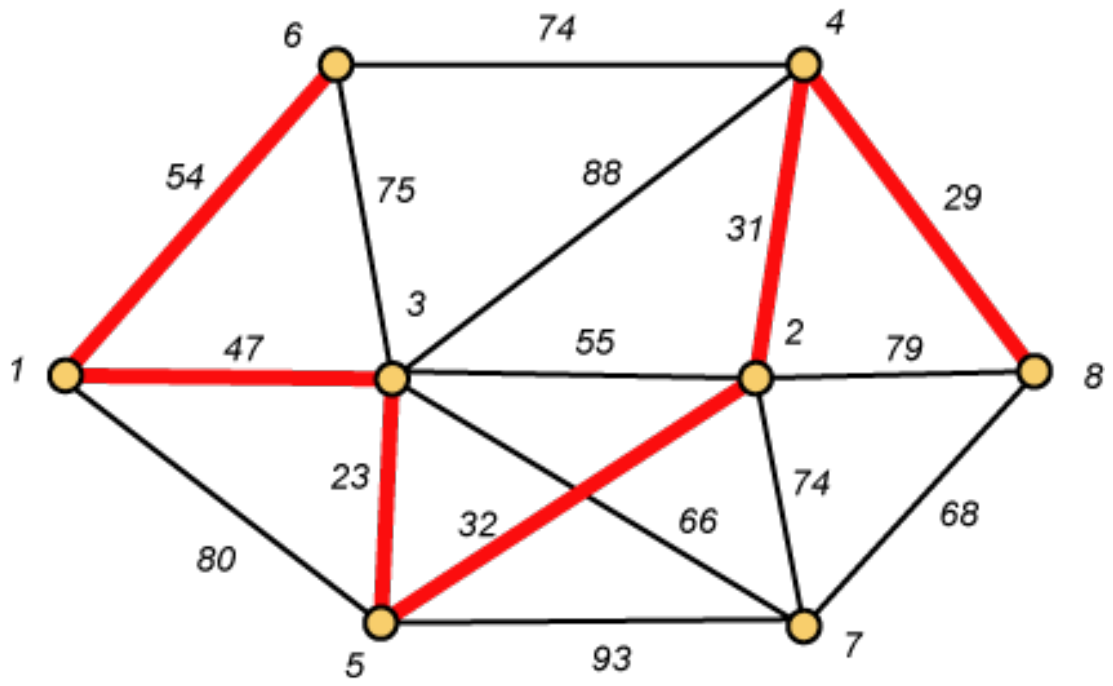
Kruskal – Step 4



Kruskal – Step 5



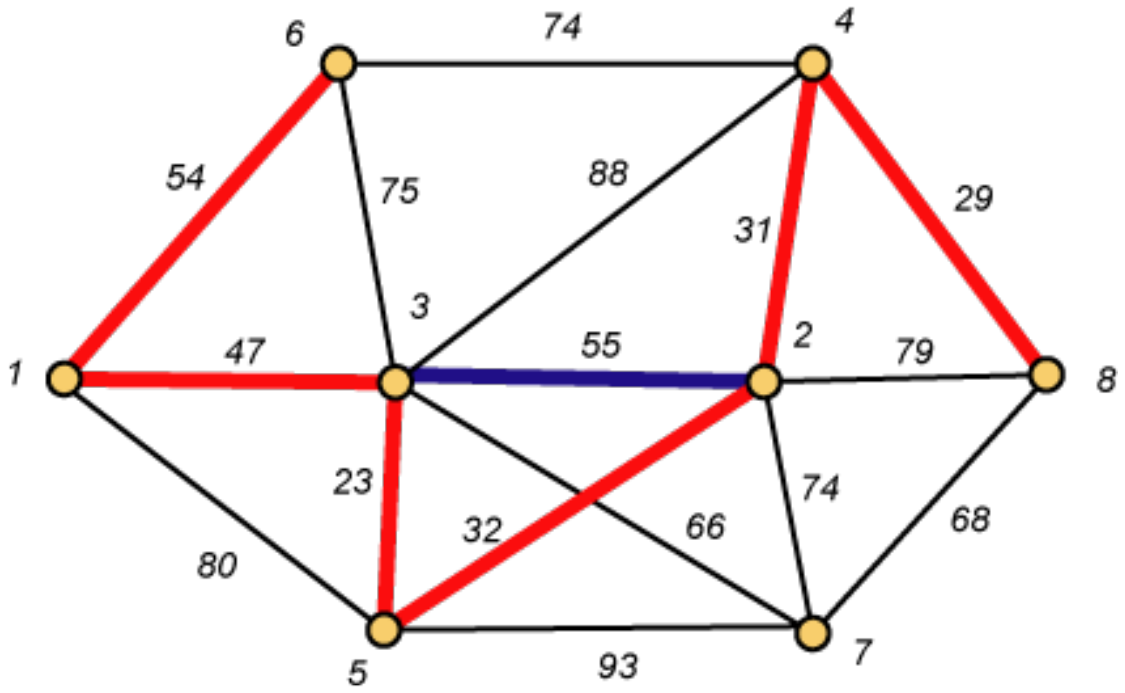
Kruskal – Step 6



Why Avoiding Cycles Matters

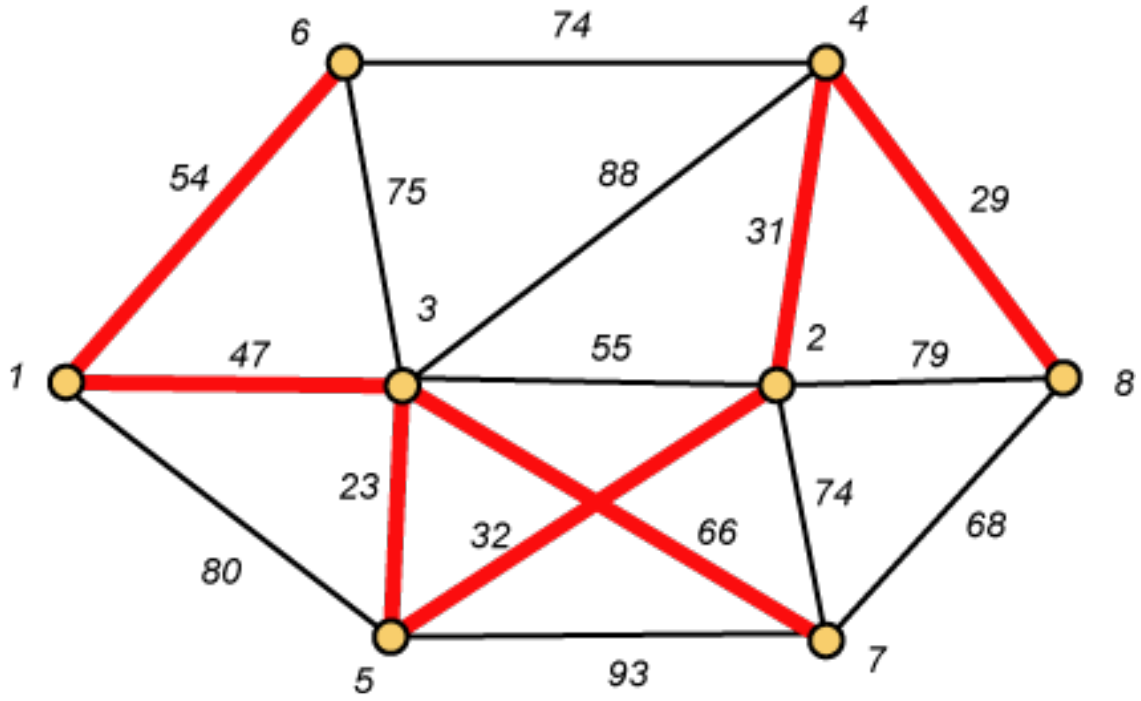
Up to this point, we have simply taken the edges in order of their weight. But now we will have to reject an edge since it forms a cycle when added to those already chosen.

Forms a Cycle



So we cannot take the blue edge having weight 55.

Kruskal – Step 7 *DONE!!*



Weight (T) = 23 + 29 + 31 + 32 + 47 + 54 + 66 = **282**

Disjoint-Set

- Keep a collection of sets S_1, S_2, .., S_k,
  - Each S_i is a set, e,g, S_1={v_1, v_2, v_8}.
- Three operations
  - Make-Set(x)-creates a new set whose only member is x.
  - Union(x, y) –unites the sets that contain x and y, say, S_x and S_y, into a new set that is the union of the two sets.
  - Find-Set(x)-returns a pointer to the representative of the set containing x.

Kruskal's Algorithm

- The algorithm adds the cheapest edge that connects two trees of the forest

```
MST-Kruskal(G,w)
01 A ← ∅
02 for each vertex v ∈ V[G] do                          O(V)
03    Make-Set(v)
04 sort the edges of E by non-decreasing weight w       O(ElogE)
05 for each edge (u,v) ∈ E, in order by non-            O(E)
   decreasing weight do
06    if Find-Set(u) ≠ Find-Set(v) then
07       A ← A ∪ {(u,v)}                                O(V)
08       Union(u,v)
09 return A
```

Overall Complexity: O(VE)

Question

- Exercise
  - 23.2.7 – Add new vertices & edges in a graph
  - 23-1: Second best minimum spanning tree