* Set Operations
* Kruskal’s Algorithm
* Prim’s Algorithm
* Minimum Spanning Tree
  + Weighted graph, say G.
  + A sub-graph T of G which avoids any cycle is a **spanning tree**
  + **weight of T** is sum of weights of all edges in T
* Greedy Approach
  + At each step make the best choice that is available at that time
    - Not looking forward or backward, just best at the moment.
  + **Does not** find globally optimum solutions

**Generic MST**

* + - Makes locally optimal choices:
      * Sort the edges by weight
      * For each edge, pick the edge with the lowest weight, that doesn’t make a cycle.
      * Step 1 takes O(m log m) = O(m log n)
      * Step 2 takes O(n log n) time
* Set Operations
  + Make-set(v) : creates a set containing element v, {v}
  + Find-set (u) : returns the set to which U belongs to
  + Union(u,v) : creates a set which is the union of the two sets, one containing v and one containing u.
* Example