# Minimum Spanning Tree - algorithms and examples in graphs

- A minimum spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles (a spanning tree) and with the minimum possible total edge weight.

## Kruskal's algorithm

- It is a greedy algorithm for finding a minimum spanning tree.
- The steps are:
  1. Create a forest (a set of trees) initially consisting of a separate single-vertex tree for each vertex in the input graph.
  2. Sort the graph edges by weight.
  3. Loop through the edges of the graph, in ascending sorted order by their weight. For each edge:
     a. Test whether adding the edge to the current forest would create a cycle.
     b. If not, add the edge to the forest, combining two trees into a single tree.
- At the termination of the algorithm, the forest forms a minimum spanning forest of the graph. If the graph is connected, the forest has a single component and forms a minimum spanning tree.
- This algorithm has the complexity of $O(|E| \log |E|)$

## Prim's algorithm

- It is a greedy algorithm for finding a minimum spanning tree.

- The steps are:
  1. For each vertex $v$ store the cheapest cost of connection to $v$, $C[v]$ and the edge providing that connection $E[v]$. By default all $C$'s are $+\infty$.
  2. Initialize the empty forest $F$ and a set of unused vertices $Q$.
  3. Repeat the following steps until $Q$ is empty:
     a. Find and remove a vertex $v$ from $Q$ having the minimum value of $C[v]$. Add that vertex to $F$.
     b. Loop over the edges $(v, w)$ connecting to $v$. For each edge, if $w$ still belongs to $Q$ and $(v, w)$ has smaller weight than $C[w]$ then set $C[w]$ to the cost of $(v, w)$ and set $E[w]$ to $(v, w)$.
  4. Return $F$ along with the corresponding edges in $E$.
- It has the time complexity of $(|V|^2)$