DT8014 Algorithms (2016)
Week 2 - Algorithm Complexity and Graphs
Exercise 2
September 7, 2016

Minimum Spanning Tree

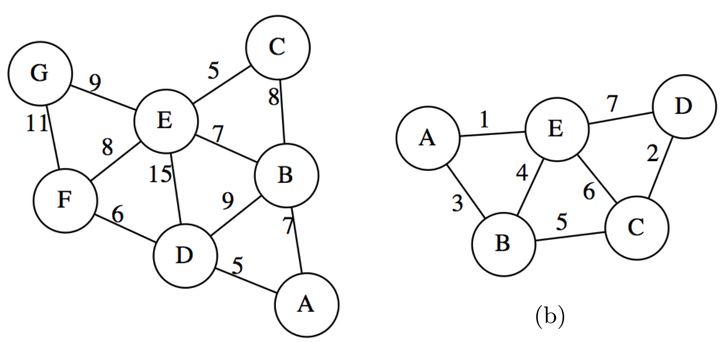
Given a connected, undirected graph, a **spanning tree** of that graph is a subgraph that is a tree and connects all the vertices together. A given graph can have many different spanning trees. We can also assign a weight to each edge, which is a number representing how unfavourable it is, and use this to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A **minimum spanning tree (MST)** or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree.

1. Describe two real-world problems which can be modeled such that an MST solves them.
2. Sketch a proof for the uniqueness property: *If each edge has a distinct weight, then the minimum spanning tree is unique.*
3. Sketch a naïve brute-force approach for computing the MST.
4. Apply Kruskal's algorithm, given in Algorithm 1, to find the MSTs for the graphs in Figure 1.
5. Demonstrate the best and worst cases of the algorithm and find the run-time of the algorithm (Big-O notation).

Algorithm 1 Kruskals Algorithm

Input: an undirected weighted graph $G = (V, E)$
Relies on a disjoint-set data structure:
- MakeSet(v): makes a set $\{v\}$ containing a single element v.
- FindSet(v): returns the set to which v belongs.
- Union(u, v) joins the subsets to which two given elements belong.

$A \leftarrow \emptyset$
for all $v \in V$ **do**
    MakeSet(v)
end for
$S \leftarrow$ all edges $(u, v) \in E$ sorted by increasing weight.
for all $(u, v) \in S$ **do**
    if FindSet(u) $\neq$ FindSet(v) **then**
        $A \leftarrow A \cup (u, v)$
        Union(u, v)
    end if
end for
return A



(a)

Figure 1: Example weighted undirected graphs.