CS 218 Design and Analysis of Algorithms

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Module 1: Basics of algorithms

Minimum Spanning Subgraph

Problem description

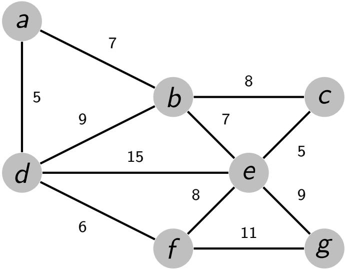
- Given an undirected connected graph $G = (V, E)$ and a cost function on the edges $c : E \to \mathbb{Z}^+$.
- Find a subset $T \subseteq E$ such that

  T must span all the vertices,

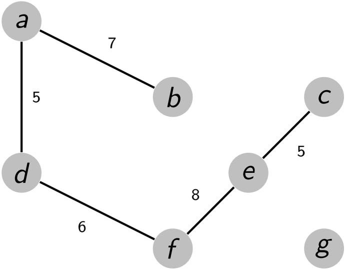
  T must be connected,

  T must be the least cost such set.

Graph G with edge costs



non-example for T



Minimum Spanning Subgraph

Problem description

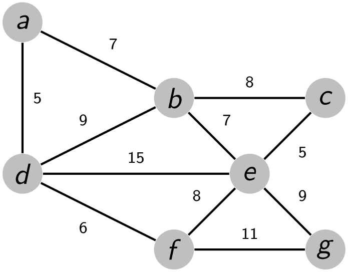
- Given an undirected connected graph $G = (V, E)$ and a cost function on the edges $c : E \to \mathbb{Z}^+$.
- Find a subset $T \subseteq E$ such that

  T must span all the vertices,

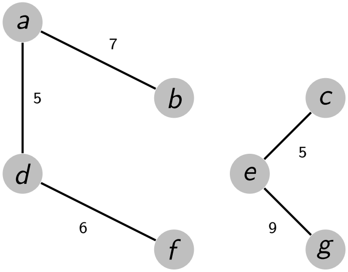
  T must be connected,

  T must be the least cost such set.

Graph G with edge costs



non-example for T



Minimum Spanning Subgraph

Problem description

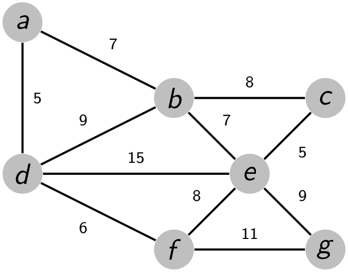
- Given an undirected connected graph $G = (V, E)$ and a cost function on the edges $c : E \to \mathbb{Z}^+$.
- Find a subset $T \subseteq E$ such that

  T must span all the vertices,

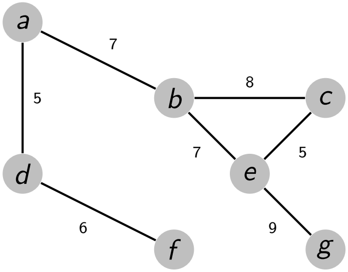
  T must be connected,

  T must be the least cost such set.

Graph G with edge costs



non-example for T



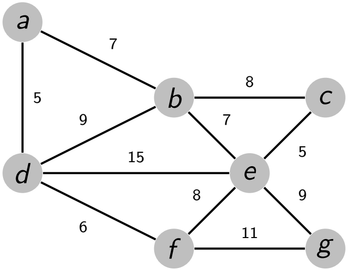
Minimum Spanning Subgraph

Problem description

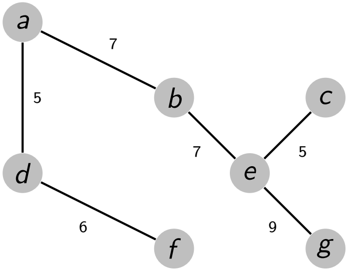
- Given an undirected connected graph $G = (V, E)$ and a cost function on the edges $c : E \to \mathbb{Z}^+$.
- Find a subset $T \subseteq E$ such that

  T must span all the vertices,
  T must be connected,
  T must be the least cost such set.

Graph G with edge costs



Example of an MST T



Minimum Spanning Tree (MST): T is a tree

Lemma

The subgraph T is a tree.

Suppose T is connected and spanning and has a cycle C.

Let e be the edge with the largest cost on the cycle C.

Suppose we remove e from T, then T still stays connected and spanning.

The weight strictly goes down.

Note that we used that the weights are positive in the above argument.

Finding a Minimum Spanning Tree (MST)

Brute-force approach

- Find distinct trees in the graph (using some standard algorithms).
- Maintain the weight of the minimum among these.

This does not work.

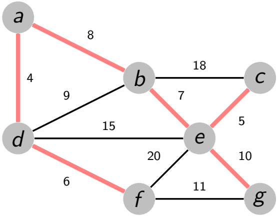
There can be exponentially many spanning trees in a graph!

We will assume that all edge costs are distinct.

Greedy approaches for MST

Greedy approach I – Kruskal's algorithm

Let $E' = \langle e_1, e_2, \ldots, e_m \rangle$, s.t. $\forall i < j$ in $[m]$, $c(e_i) < c(e_j)$
$\{$ E' is the array of edges sorted in the increasing order of their cost. $\}$
$T \leftarrow \varnothing$, $i \leftarrow 1$.
while $i \le m$ **do**
    if $T \cup \{e_i\}$ does not have a cycle **then**
        $T \leftarrow T \cup \{e_i\}$
    else
        $i \leftarrow i + 1$
    end if
end while
Output T



Correctness of Kruskal's algorithm

To argue about the correctness of Kruskal's algorithm we need to show

- The subgraph T computed by the algorithm does not have cycles.
- T is connected.
- T is a minimum spanning tree.

By the design of the algorithm T does not have cycles.

To prove the rest, we need to make a graph-theoretical observation about minimum spanning trees.

Correctness of Kruskal's algorithm

Lemma (The cut property)

Let S be any non-empty strict subset of V. Let $e = (v, w)$ be the minimum cost edge such that $v \in S$ and $w \in V \setminus S$. Then every minimum spanning tree must contain e.

Correctness of Kruskal's algorithm

Correctness of Kruskal's algorithm.

✓ The subgraph T computed by the algorithm does not have cycles.
By the design of the algorithm T does not have cycles.

- T is connected.
- T is a minimum spanning tree.

To prove these, we will use the Cut Property.