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

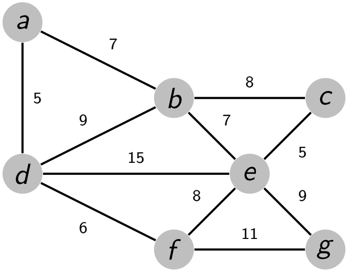
- Given an undirected connected graph $G = (V, E)$ and a cost function on the edges $c : E \to \mathbb{Z}^+$.
- Find a subset $T \subseteq E$ such that

  T must span all the vertices,

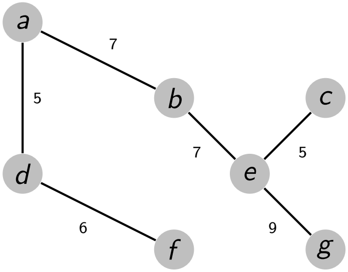
  T must be connected,

  T must be the least cost such set.

Graph G with edge costs



Example of an MST T



Greedy approaches for MST

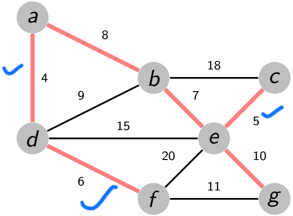
Greedy approach I – Kruskal's algorithm

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. }
$T \leftarrow \emptyset$, $i \leftarrow 1$.
while $i \leq m$ **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

c S

By cut property

d-f is certainly be present in MST



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

Any vertex outside needs to be covered

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.

Then because of connectedness, there will be atleast one such edge e. Min^m one should be in our MST

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.

Correctness of Kruskal's Algorithm

When we are selecting min^m edge.

T is a minimum spanning tree.

Let T' be the set created by algorithm at some intermediate step.

Let $e = (v, w)$ be the first edge added by the algorithm to T' during one of the subsequent steps.

Let S be the set of all the nodes that v is connected to in T'.

only consider edges in T'

As the algorithm could add e to T', ~~implies~~ $\iff$ there was no edge in T' connecting any node in S to any node in $V \setminus S$.

From the working of the algorithm, we know that e must be the lowest cost such edge.

But by the Cut Property, we know that e must be present in the final minimum spanning tree T.

Thus the algorithm makes correct choices at each step.

Correctness of Kruskal's Algorithm

T is connected.

Suppose it is not connected.

There is a non-empty set S such that no edge from S to $V \setminus S$ in T.

As G itself is connected, there is some edge that connects S to $V \setminus S$.

By the Cut Property the minimum cost such edge must be in T.

Therefore it contradicts our assumption.

Correctness of Kruskal's algorithm

Correctness of Kruskal's algorithm.

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

To prove these, we used the Cut Property.

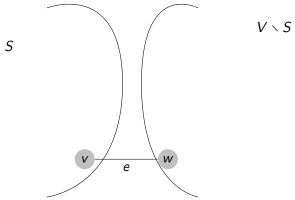
Correctness of Kruskal's algorithm

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.

Proof (by the exchange argument)

Let S and e be as above.



Let T be such that it is a spanning tree and it does not contain e.

Then adding e to it creates a cycle.

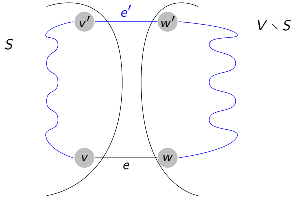
Correctness of Kruskal's algorithm

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.

Proof (by the exchange argument)

We know that v and w are connected in T, say through path P. Let v' be the last vertex along this path in S and w' be the first vertex in $V \setminus S$.



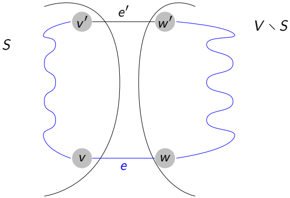
Correctness of Kruskal's algorithm

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.

Proof (by the exchange argument)

We know that v and w are connected in T, say through path P. Let v' be the last vertex along this path in S and w' be the first vertex in $V \setminus S$.



Let $T' \leftarrow T \setminus \{e'\} \cup \{e\}$. T' has lower cost than T.

Another Greedy Approach for finding MST

Maintaining a connected tree.

Start with an arbitrary vertex.

In each iteration add the edge with the smallest cost among the edges leaving the current set of vertices.

Repeat until no more edges can be added.

Note that this uses the Cut Property in a very direct way.