W3U2: Krushkal's Algorithm

C Pandu Rangan

Recap

- Prim's Algorithm

Outcomes

- Krushkal's Algorithm

Krushkal's Algorithm

- Krushkal's algorithm is a greedy algorithm designed somewhat 'directly and obviously' based on the definitions of trees and cost of trees.
- The idea is to construct a spanning tree with as many cheap edges as possible. Thus, we may start with a list L of all edges sorted in the increasing order of their weights and consider the edges of L one by one in that order for a possible inclusion in the tree. This way, cheaper edges are considered earlier and costlier edges are considered later. It is easy to visualize and express our plan as an iterative algorithm. In each iteration of the algorithm we pick an edge and decide if we should add it to T or reject it. Thus, T consists of a 'growing' set of edges and the hope is in the end we have a minimum spanning tree.

Krushkal's Algorithm (contd)

Our rejection rule is also simple and direct.
If an edge e in L forms a cycle with some of the edges included in T, we cannot include e in T as T is supposed to be an acyclic structure. Hence, we reject e and proceed with the next edge in L.

Outline

We present out idea neatly in the pseudocode below:

\\G $= (V, E)$ is a connected, weighted, undirected graph. Weight of an edge e is denoted by $w(e)$.

\\L is the list of edges of $\underline{E}$ in increasing order of their weights.

\\T $= (V, A)$ is the output.

\\A is the set of edges selected by the algorithm.

1) $A = \emptyset; e = first\ edge\ of\ L;$

2) while (NOT end of L)

2.1) if (e does not form a cycle with any subset of edges in A)

$A = A \cup e.$

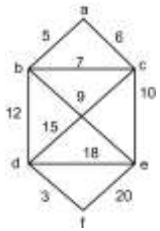
2.2) $e = Next\ (e, L).$

Note

At this point it is not even clear why the set of edges in A form a tree. Even if A forms a spanning tree, it is not obvious that it forms a minimum spanning tree. We will see the details related to these questions shortly.

Example

The following example should give a clear understanding on the working of the algorithm.

Let $G =$ 

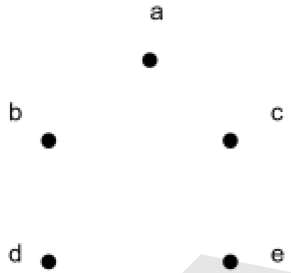
The sorted list of edges L is

$$\left\{\begin{matrix}(d,f),(a,b),(a,c),(b,c),(b,e),\\ (c,e),(b,d),(c,d),(d,c),(e,f)\end{matrix}\right\}$$

Example (contd)

Initially, since A is empty, T may be represented by the set of vertices $\{a, b, c, d, e, f\}$ without any edges as shown below:

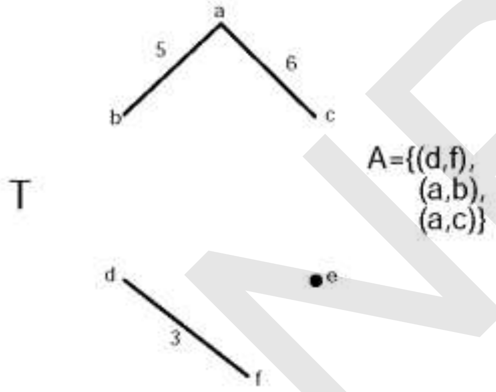
T



$A = \emptyset$

Example (contd)

In the first three iterations, we could include the first three edges of L in A as they do not form any cycle with previously included edges. The graph T with edges in A at this stage is shown below.

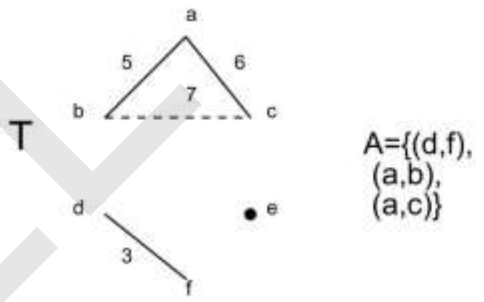


Example (contd)

In the next iteration, the fourth edge of L, namely (b, c), will be considered. However, the edge (b, c) will be ignored because (b, c) forms a cycle with edges (a, b) and (a, c) which are in A. This is shown with (b, c) in dotted line in the picture.

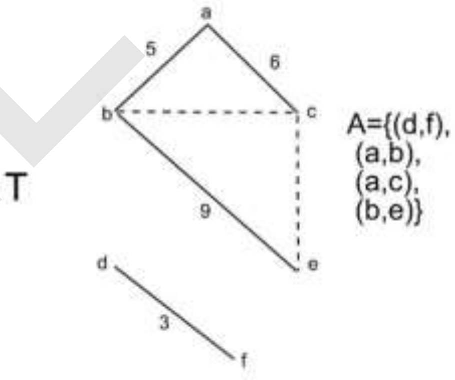
Considered, however, the edge (b, c) will be ignored because (b, c) forms a cycle with edges (a, b) and (a, c) which are in A. This is shown with (b, c) in dotted line in the picture.

Edges represented in dotted lines are ignored edges.



Example (contd)

In the next couple of iterations, the edge (b, c) will be added and the edge (c, e) will be rejected as shown.



Note that (c, e) is ignored because it forms a cycle with $\{(c, a), (a, b), (b, e)\}$

Continuing with the example, in the next iteration the edge (b, d) is picked up and included in A and all further edges are rejected.

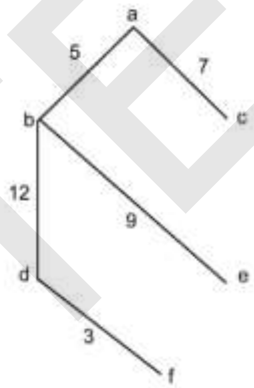
Example (contd)

$A = \{(d,f),(a,b),(a,c),(b,e),(b,d)\}$

T is indeed a spanning tree and its cost is

$3 + 5 + 7 + 9 + 12 = 36.$

T



We will now prove that correctness of the algorithm. This algorithm was designed by Krushkal and we refer the same as Krushkal Algorithm for MST.

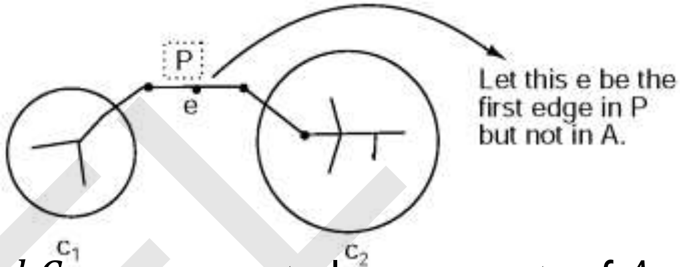
Proof

We first prove that the set of edges in A form a tree. The graph induced by A is clearly acyclic. We need to prove that A induces a connected graph. If A induces a disconnected graph, there would exist an edge in E that would not form a cycle with any subset of edges in A. For example any edge that is not in A but in a path connecting two vertices in two different components of A will have this property.

Proof (contd)

P is a path in G, connecting two vertices from two different components. e is an edge in P but $e \notin A$.

1. P is a path in G, connecting two vertices from two different components. e is an edge in P but $e \notin A$.
2. Let this e be the first edge in P but not in A.
3. This will not form a cycle with any set of edges in A.

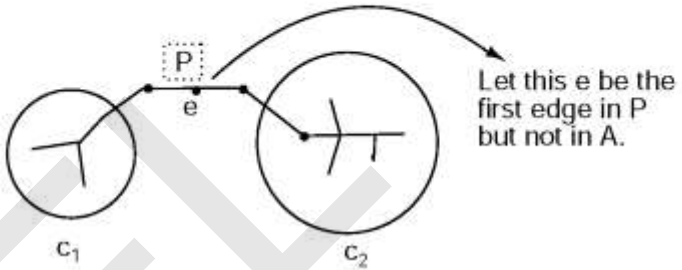


C_1 *and* C_2 are connected components of A

Proof (contd)

4. If it forms, it would define a path to a vertex outline this component
5. $e \notin A$ is a contradiction because when e was examined by the algorithm e would have been included in A.

Thus, A induces a connected acyclic graph and this means A forms a tree.
Thus, A contains exactly $(n-1)$ edges, where $n = |v|$



C_1 *and* C_2 are connected components of A

Thank You