Trees

Pushpendra Kumar Pateriya
School of Computing & Information Technology
Lovely Faculty of Technology & Sciences
Lovely Professional University
E-mail: pushpendra.14623@lpu.co.in

September 17, 2013

Outline

1. Forest and Trees
2. Rooted trees and Branchings
3. Minimum Spanning Tree
   - Kruskal's algorithm
   - Prim's algorithm

Forest and Trees I

Definitions

- A connected acyclic graph is called a **tree**.
- An acyclic graph is called a **forest**.
  Thus, every component of a forest is a tree.
- **leaf of the tree**: vertex of degree exactly one.

Forest and Trees II

Theorem

If T is a tree, then $e(T) = v(T) - 1$.

Theorem

graph G is a tree if and only if any two vertices are connected by a unique path.

Theorem

A connected graph G is a tree if and only if every edge is a cut-edge.

Rooted trees and Branchings I

- A *rooted tree* T(x) is a tree T with a specified vertex x, called the root of T.
- An orientation of a rooted tree in which every vertex but the root has indegree one is called a *branching*.

Theorem

For any graph G, the following statements are equivalent.

(a) *G is a tree.*

(b) *G is connected and $m = n - 1$.*

(c) *G is acyclic and $m = n - 1$.*

Number of trees I

Question

How many trees are there on a given set of n vertices, where two trees T_1 and T_2 are counted distinct if $E(T_1) \neq E(T_2)$?

Theorem

(**Cayley's formula** *for the number of distinct trees). The number of distinct trees on n vertices is* n^{n-2}.

Number of trees II

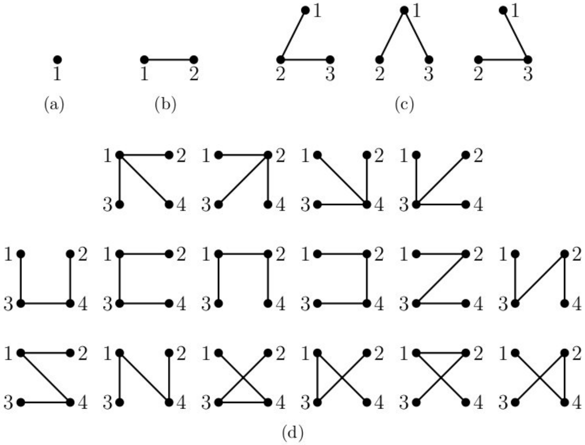
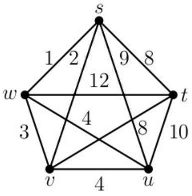


Figure: Distinct trees upto four vertices.

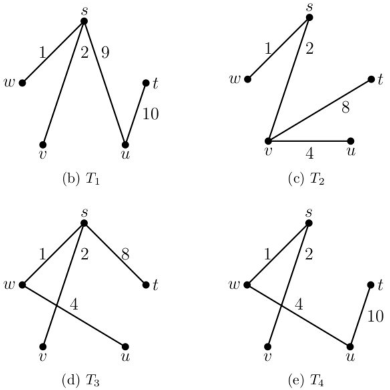
Minimum Spanning Tree I

A spanning tree of minimum weight in a connected weighted graph G is called a **minimum spanning tree** of G.



(a) G

Minimum Spanning Tree II



(b) T_1 (c) T_2 (d) T_3 (e) T_4

Figure: A weighted graph G and four of its spanning trees T_1, T_2, T_3, and T_4, with $W(T_1) = 22$, $W(T_2) = 15$, $W(T_3) = 15$ and $W(T_4) = 17$.

Minimum Spanning Tree III

Two Algorithms to find a MST of a connected weighted graph G.

1. **Kruskal's algorithm (J.B. Kruskal, 1956)**
   Input: A weighted connected graph (G, W)
   Output: A minimum spanning tree T_k of G.
   Step1(Initial): Arrange the edges of G in non-decreasing order, say π. Define H_0 to be the graph with vertex set V (G) and no edges.
   Step 2: Select the first edge from π say e_1 and add it to H_0. Call the resulting graph H_1.
   Step 3 (Recursion): After selecting i edges $e_1, e_2, ..., e_i$ and forming the graph H_i, find the next edge, say e_{i+1}, in the subsequence succeeding e_i in π which does not create a cycle with the edges already selected. Add it to H_i and call the resulting graph H_{i+1}.

Minimum Spanning Tree IV

Step 4 (Stop Rule): Stop when n − 1 edges are selected. Output H_{n-1} as T_k.

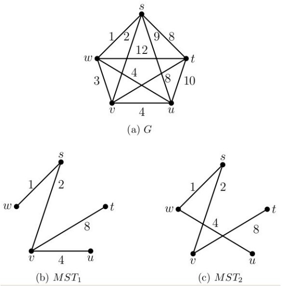


Figure: Two minimum spanning trees of G constructed by Kruskal's algorithm. Notice that both the trees have the same weight 15 but they are not isomorphic.

Minimum Spanning Tree V

2. **Prim's algorithm (R. C. Prim, 1957)**
   Input: A weighted connected graph (G, W)
   Output: A minimum spanning tree Tp of G.
   Step 1: Select a vertex v_1 (arbitrarily). Select an edge of minimum weight in $[\{v_1\}, V - \{v_1\}]$, say (v_1, v_2). Define H_1 to be the tree with vertices v_1, v_2 and the edge (v_1, v_2).
   Step 2: Having selected the vertices, $v_1, v_2, ..., v_k$ and k − 1 edges, and forming the tree H_{k-1}, select an edge of minimum weight in $[\{v_1, ..., v_k\}, V - \{v_1, ..., v_k\}]$, say (v_i, v_{k+1}). Define H_k to be the tree obtained by joining v_{k+1} to $v_i \in H_{k-1}$.
   Step 3 (Termination): Stop when n − 1 edges are selected and output H_{n-1} as T_p.

Minimum Spanning Tree VI

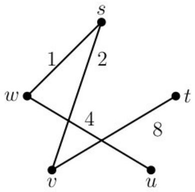


Figure: MST constructed using Prim's algorithm