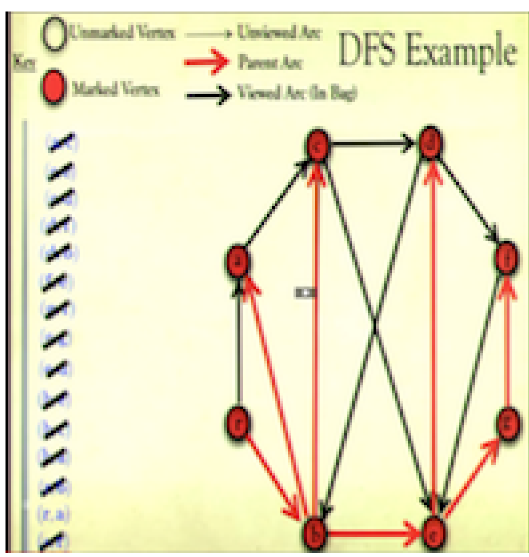
Lecture 9: Graph Algorithms: Depth First Search

1 Depth First Search: Stack

- Using a **Stack** (LIFO) data structure produces:
  Depth First Search Algorithm
  Add (*, r) to a Stack
  While the Stack is non-empty
  Remove the first arc (u, v) from the Stack
  If v is unmarked
  Mark v
  Set $p(v) \leftarrow u$
  For each arc(v,w)
  Add (v, w) into the back of the Stack



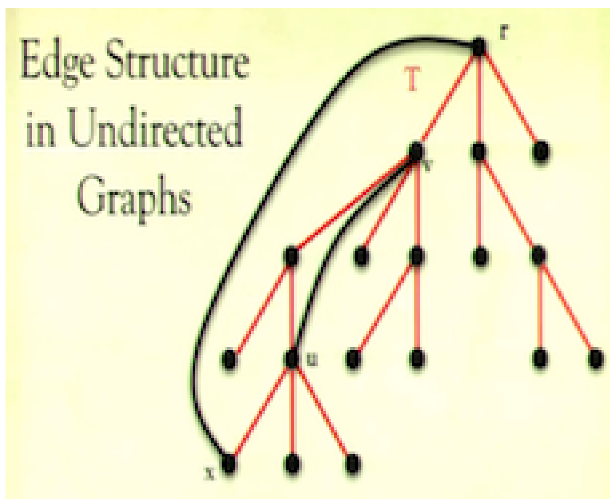
2 Depth First Search Trees

- Recall, we proved that the search algorithm produces a search tree.
  Theorem. Let G be a connected, undirected graph. Then the predecessor edges form a tree rooted at r.

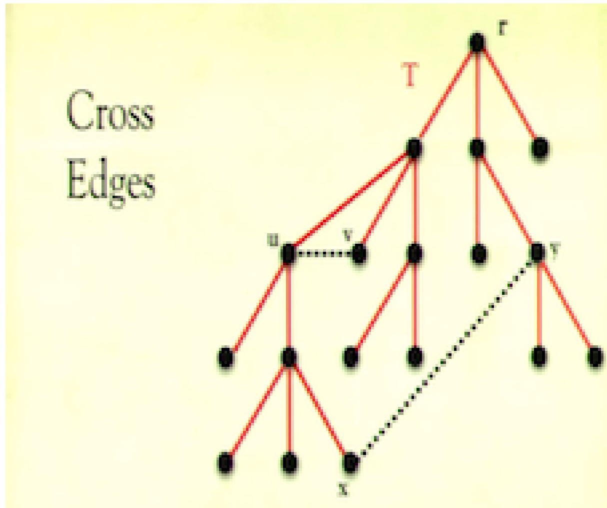
- However, the structure of the DFS tree is quite different from that of a BFS tree.

3 Edge Structure in Undirected Graphs

- Depth First Search partitions the edges of an undirected graph into two types:
  Tree Edges. Predecessor edges in the DFS tree T.
  Back Edges. Edges where one endpoint is an ancestor of the other endpoint in T.



- When we perform a DFS, there is no crosses of the following type:
  Cross Edges. Edges where neither endpoint is an ancestor of the other in T.



4 Depth First Search (Recursive Description)

- The Depth First Search algorithm can be defined recursively:

  RecursiveDFS(r)
  Mark r
  For each edge (r, v)
    If v is unmarked
      Mark v
      For each arc (r, v)
        Set $p(v) \leftarrow r$
        RecursiveDFS(v)

5 Depth First Search (Recursive Description)

Theorem. Let T be a DFS tree in an undirected graph G. Then, for every edge (u, v). wither u is an ancestor of v in T or v is an ancestor of u.
Proof.

- Wlog assume u is marked before v.
- Consider the time u is marked during **RecursiveDFS**(u)
- In **RecursiveDFS**(u) the algorithm examines each arc incident to u.

<u>Case 1:</u> v is unmarked when **RecursiveDFS**(u) examines (u, v).

- The **RecursiveDFS**(u) sets $p(v) \leftarrow u$
    - (u, v) is a tree edge.

<u>Case 2:</u> v is marked when **RecursiveDFS**(u) examines (u, v).

- But v was marked after u, so it was marked during **RecursiveDFS**(u)
- Thus, we have a series of vertices $\{u = w_0, w_1, ..., w_{l-1}, w_l = v\}$ where $p(w_k) = w_{k-1}$
    - u is an ancestor of v in T.
    - (u, v) is a back edge.