Applications of DFS in Directed Graphs

Types of Edges

- In DFS, first we have to identify the ***tree edges***
  - Using the ***tree edges*** you can identify the parent child relationships
  - And the ancestors descendant relationships
  - Using that you can figure out whether the other edges are ***forward edge*** or a ***back edge*** or a ***cross edge***
    - If the two end points have an ancestor descendant relationship, then it's either a ***forward edge*** or a ***back edge*** (apart from a ***tree edge***)
      - If the starting vertex is an ancestor of the ending vertex: ***forward edge***
      - If the starting vertex is a descendant of the ending vertex: ***back edge***
    - Otherwise, it is a ***cross edge***

Application: Directed Graph is Cyclic using DFS

- How to check whether a directed graph is cyclic?
- How to check whether a undirected graph is cyclic?
  - If there is a back edge
  - Because the two end points of the back edge are connected in the tree

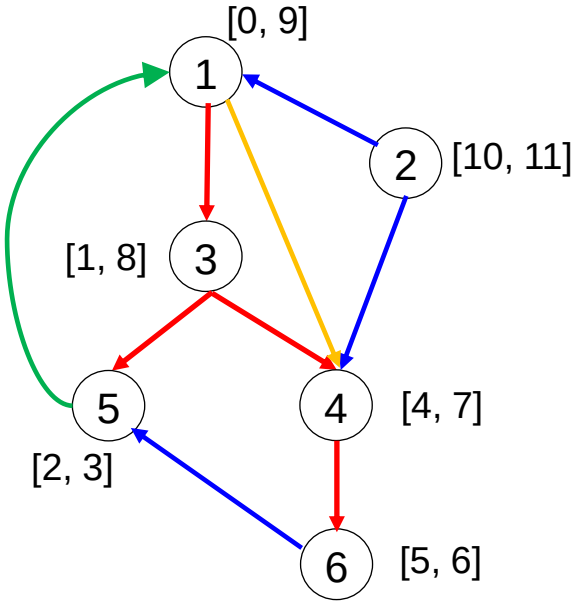
- How to check whether a directed graph is cyclic?
  - If there is a back edge
  - ***If there is no back edge, does that mean graph G is acyclic?***

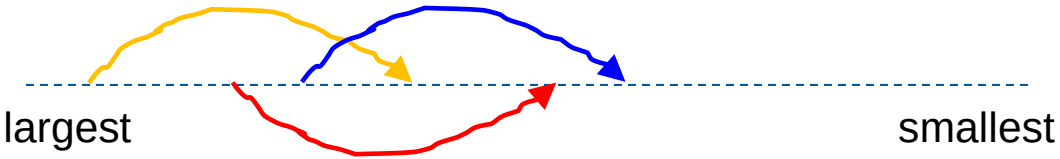
(graph with no cycle)

Application: Directed Graph is Cyclic using DFS

- To prove: ***If there is no back edge, does that mean graph G is acyclic?***

  1. Do DFS
  2. Lets order vertices by their ***departure time***





departure time →

All are going from left to right, ***hence no cycle***

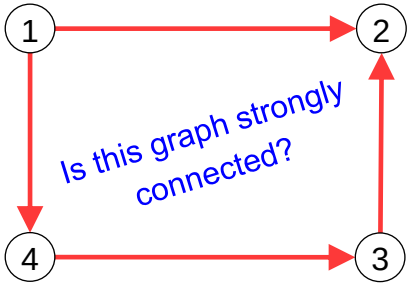
Application: Topological Sort

- Given an acyclic graph, lets order the vertices of the graph according to decreasing departure times
- Then every edge goes from the left to right
- *Thus, in an acyclic graph, we can linearly order the vertices so that the edges are going only from left to right*
- *This ordering is called the* **topological sort**
- Time complexity: ***O(V + E)***
- ***Question:*** What is minimum and maximum departure time we can have?

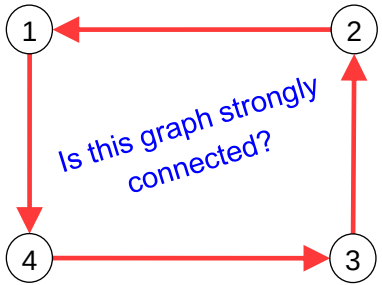
A directed graph which is acyclic is also called a ***DAG***

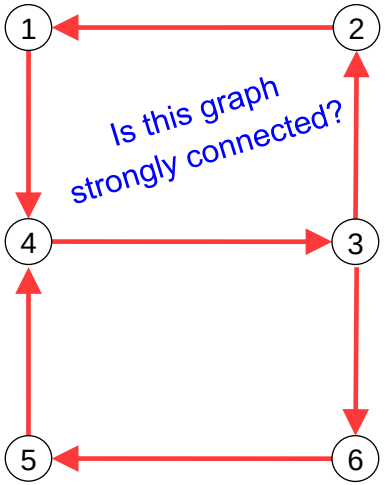
Strongly Connected/Weakly Connected Graph

- When is an undirected graph connected?
  - If there is a path between every two vertices
- When is an directed graph connected?
  - A directed graph is called ***strongly connected***, if there is a path between every ordered pair of vertices



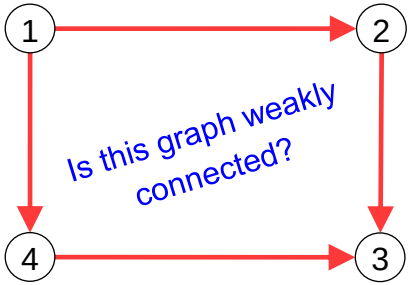
There is a path from ***1*** to ***2*** but there is no path from ***2*** to ***1***
There is a path from ***3*** to ***2*** but there is no path from ***2*** to ***3***
There is a path from ***4*** to ***3*** but there is no path from ***3*** to ***4***



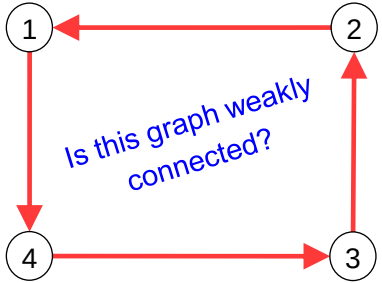


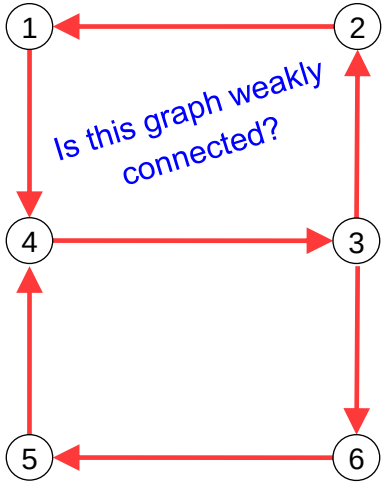
Strongly Connected/Weakly Connected Graph

- When is an directed graph ***weakly connected***?
  - A directed graph is called ***weakly connected***, if for every pair of vertices ***u*** and ***v***, there is a path from ***u*** to ***v*** or ***v*** to ***u*** or both



There is no path from ***2*** to ***4*** and also no path from ***4*** to ***2***



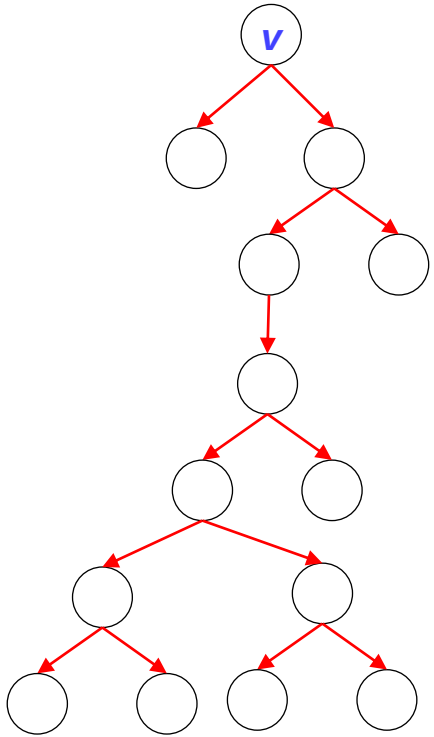


Application: Graph is Strongly Connected using DFS

- Given a graph ***G***, is ***G strongly connected***?
  - Start DFS from any vertex, and it should visit all the vertices
  - Take the next vertex, do a DFS, and it should visit all the vertices
  - Take the next vertex, do a DFS, and it should visit all the vertices
  - And so on…
  - If in each one of these DFS's, we include all the vertices of the graph only then is the graph strongly connected
  - Time complexity: ***O(VE)***
  - Can we do better? Can we do it in linear time? ***O(V + E)***

Application: Graph is Strongly Connected using DFS

- What happens after first DFS(***v***)? ***v***: random starting vertex
  - If DFS(***v***) visits all vertices in graph ***G*** then there exists a path from ***v*** to every vertex in ***G***
  - Lets' assume: there exists a path from every vertex in ***G*** to ***v***
  - Does that together imply that the graph is strongly connected?
    - If we have to find a path between some two vertices, say ***x*** and ***y***, we go from ***x*** to ***v*** (by assumption), and then we go from ***v*** to ***y*** (property of DFS)



Next Lecture

Applications of DFS in Directed Graphs

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in, shrutilipi.bhattacharjee@tum.de