Algorithms

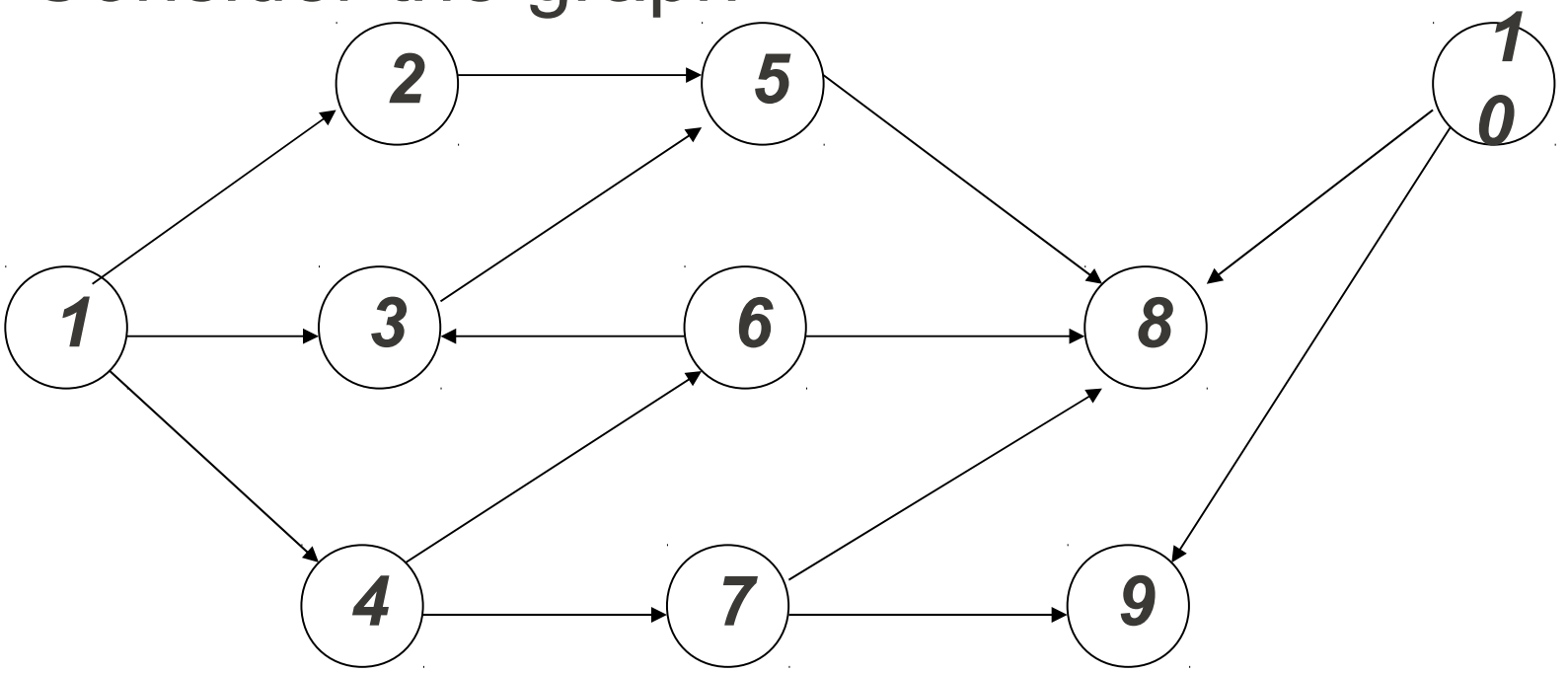
Graph Search

Graph Search Methods

- Many operations require us to visit all vertices that can be reached from a given start vertex.
- A vertex u is reachable from graph v iff there is a (directed) path from v to u.
- The 2 standard ways to search for these vertices are **breadth first search (BFS)** and **depth first search (DFS)**
- Both search methods are popular but the DFS method is used more frequently to obtain efficient graph algorithms

Breadth First Search

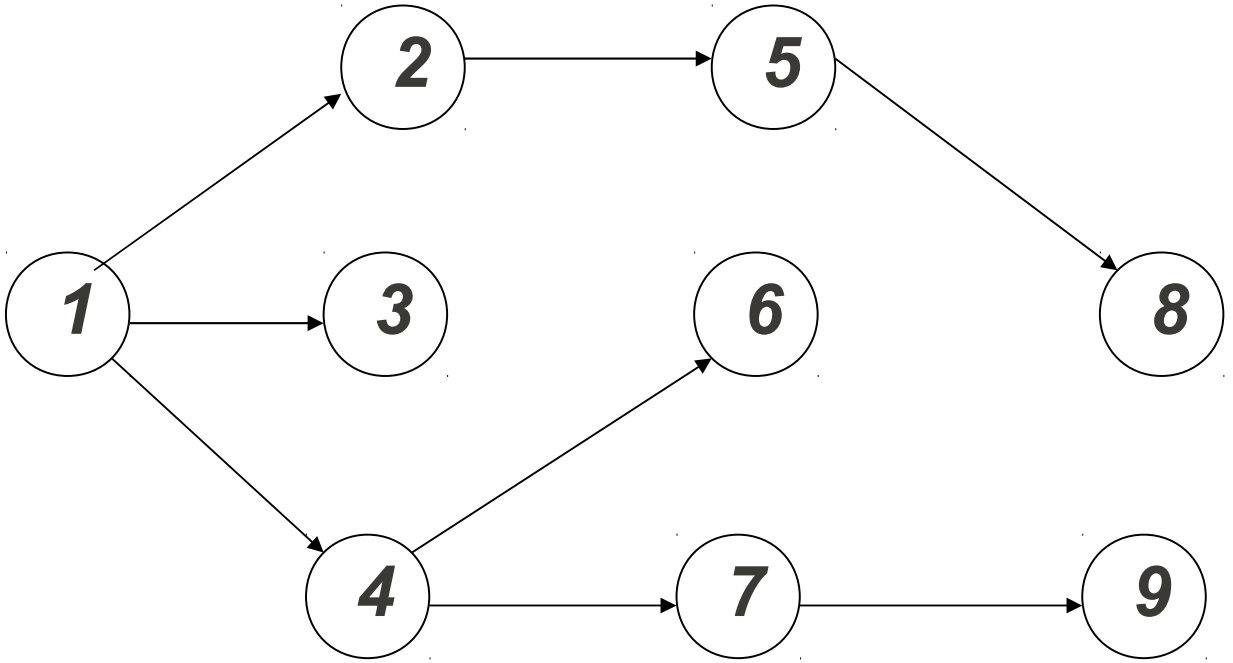
- Given a graph G = (V, E) and a distinguished **source** vertex s, breadth first search systematically explores the edges of G to ‘discover’ every vertex that is reachable from s.
- Consider the graph



Breadth First Search

- One way to determine all the vertices reachable from vertex 1 is to determine the set of vertices adjacent from 1. This set is {2, 3, 4}.
- Next we determine the set of **new** vertices (i.e. vertices not yet reached) that are adjacent from vertices in {2, 3, 4}. This set is {5, 6, 7}.
- The set of new vertices adjacent from vertices in {5, 6, 7} is {8, 9}.
- No new vertices are adjacent from a vertex in {8, 9}.
- Therefore {1, 2, 3, 4, 5, 6, 7, 8, 9} is the set of vertices reachable from vertex 1.

Breadth First Search



Depth First Search

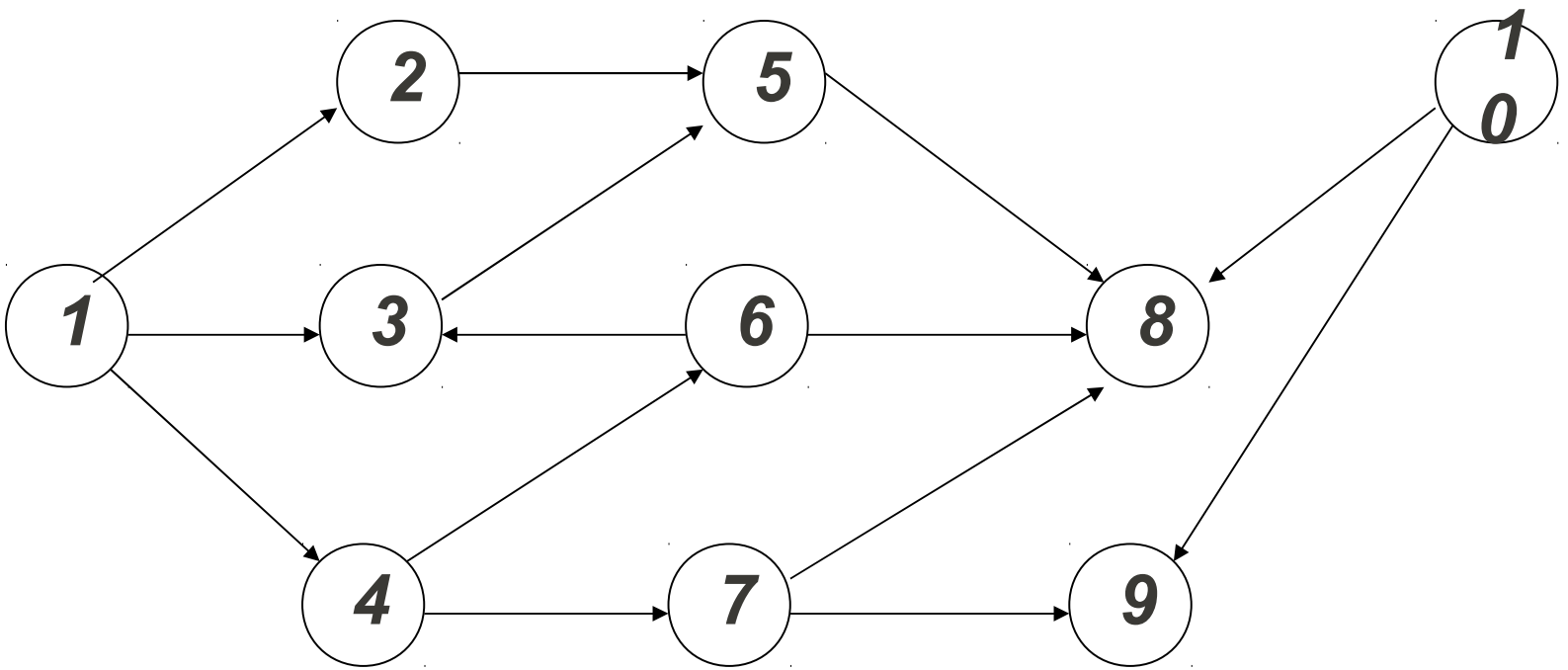
- Depth First Search is an alternative to Breadth First Search
- As the name suggests, the strategy of a DFS is to search deeper in the graph whenever possible.
- Given a starting vertex ***v***, a DFS proceeds as follows:
- First the vertex ***v*** is marked as reached, and then an unreached vertex ***u***, adjacent from ***v*** is selected.
- If such a vertex does not exist, the search terminates.

Depth First Search

- Assume that a ***u*** as described exists.
- A DFS from u is now initiated.
- When this search is completed, we select another unreached vertex adjacent from v.
- If such a vertex does not exist, then the search terminates.
- If such a vertex exists, a DFS is initiated from this vertex and so on.

Depth First Search

Consider the graph



Depth First Search

- If the starting vertex ***v*** is 1, then the candidates for ***u*** are 2, 3 and 4.
- We will choose 2 as the first value that is assigned to ***u***.
- A DFS from 2 is now initiated.
- Vertex 2 is now marked as reached
- The only candidate for ***u*** this time is vertex 5.
- A DFS from 5 is now initiated and vertex 5 is marked as reached
- The only candidate for ***u*** this time is vertex 8.

Depth First Search

- A DFS from 8 is now initiated and vertex 8 is marked as reached
- From vertex 8 there are no unreached adjacent vertices, so the algorithm backs up to vertex 5.
- There are no new candidates for **u** at vertex 5, so we back up to vertex 2.
- There are no new candidates for **u** at vertex 2, so we back up to vertex 1.
- At this point we have 2 candidates for ***u***, vertex 3 and vertex 4.

Depth First Search

- We can choose either but in this case we will choose 4 as the value for **u.**
- A DFS is now initiated from vertex 4.
- We mark 4 as reached.
- There are 3 candidates for ***u*** this time, vertices 3, 6 and 7.
- Again we can choose any of them, in this case we choose vertex 6.
- We mark 6 as reached.
- A DFS is now initiated from vertex 6.

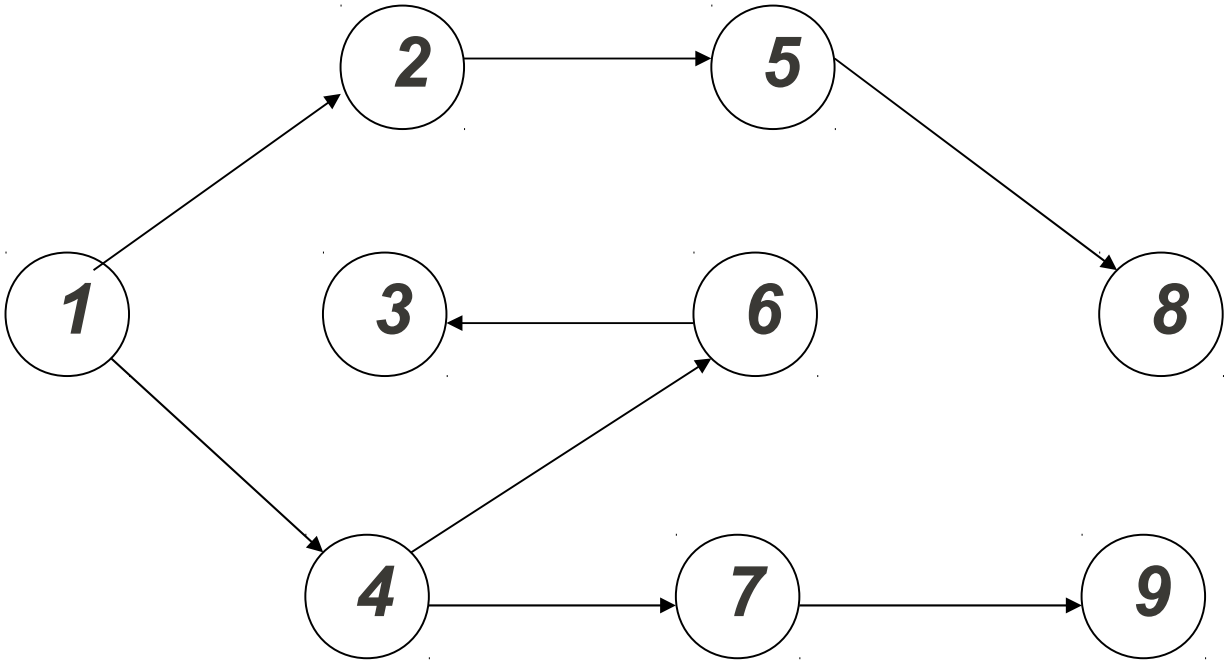
Depth First Search

- The only candidate for ***u*** this time is vertex 3.
- A DFS from 3 is now initiated and vertex 3 is marked as reached
- From vertex 3 there are no unreached adjacent vertices, so the algorithm backs up to vertex 6.
- There are no new candidates for **u** at vertex 6, so we back up to vertex 4.
- The only candidate for ***u*** this time is vertex 7.
- A DFS from 7 is now initiated and vertex 7 is marked as reached

Depth First Search

- The only candidate for ***u*** this time is vertex 9.
- A DFS from 9 is now initiated and vertex 9 is marked as reached.
- From vertex 9 there are no unreached adjacent vertices, so the algorithm backs up to vertex 7.
- There are no new candidates for **u** at vertex 7, so we back up to vertex 4.
- There are no new candidates for **u** at vertex 4, so we back up to vertex 1.
- There are no new candidates for **u** at vertex 1, so the algorithm terminates.

Depth First Search



Applications of BFS and DFS

Finding a path

We can find a path from a source vertex ***s*** to a destination vertex ***d*** by starting a search (either breadth first or depth first) at vertex **s** and terminating the search as soon as we reach vertex **d**.

Applications of BFS and DFS

Connected Graphs

We can determine whether an undirected graph G is connected by performing either a DFS or a BFS from any vertex and then verifying that all vertices have been labelled as reached. Although this strategy directly verifies only that there is a path from the start vertex of the DFS or BFS to every other graph vertex, this verification is sufficient to conclude that a path exists between every pair of vertices

Applications of BFS and DFS

Spanning Trees

If a BFS is carried out starting from any vertex in a connected undirected graph of network with n vertices, then we know that all vertices will get labelled and exactly n-1 edges is used to reach these vertices. A **breadth first spanning tree** is any spanning tree obtained from a BFS.

When a DFS is performed in a connected undirected graph or network, again exactly n-1 edges are used to reach the new vertices. A **depth first spanning tree** is any spanning tree obtained from a DFS.