Discrete Structures– Graphs: BFS and DFS

Dr. Ab Mosca (they/them)

Plan for Today

- Trees
  - Breadth first search (BFS)
  - Depth first search (DFS)
- Algorithm Analysis

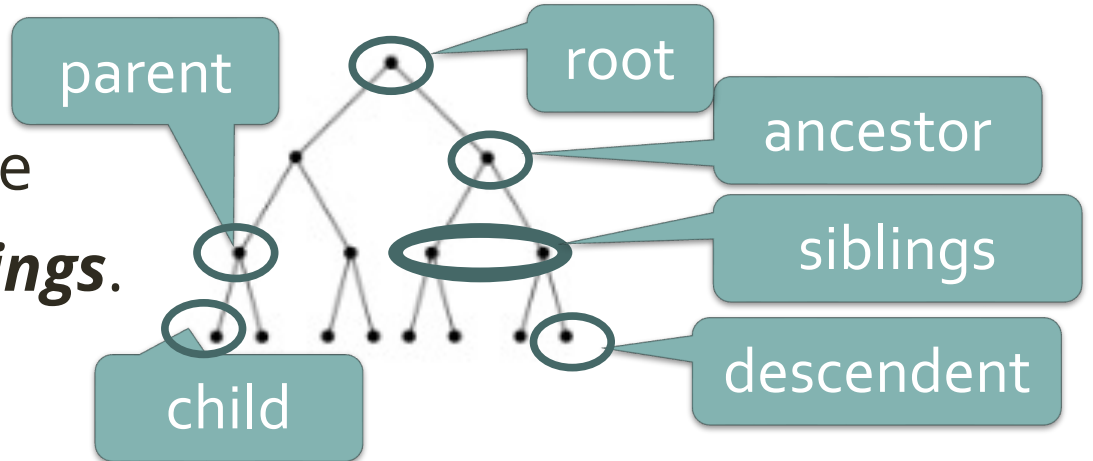
Warm Up: Rooted Trees

We can identify one vertex in a tree as the ***root***. Then, every other vertex on the tree can be characterized by its position relative to the root.

If two vertices are adjacent, we say the one closer to the root is the ***parent***, and the other is the ***child***.

In general, we say a vertex, v, is a ***descendent*** of a vertex, u, provided u is a vertex on the path from v to the root. Then, we would call u an ***ancestor*** of v.

Vertices with the same parent are called ***siblings***.



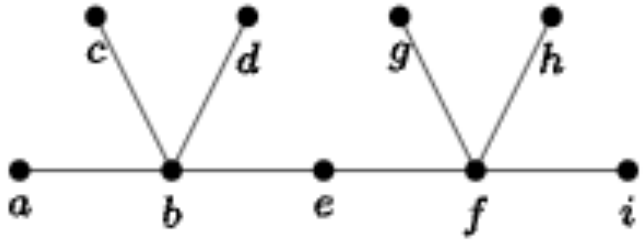
Warm Up: Rooted Trees

We can identify one vertex in a tree as the ***root***. Then, every other vertex on the tree can be characterized by its position relative to the root.

If two vertices are adjacent, we say the one closer to the root is the ***parent***, and the other is the ***child***.

In general, we say a vertex, v, is a ***descendent*** of a vertex, u, provided u is a vertex on the path from v to the root. Then, we would call u an ***ancestor*** of v.

Vertices with the same parent are called ***siblings***.



Let e be the root. Label the other vertices.

Navigating Trees

Often, we want to visit each vertex in a tree, or find a specific vertex. Because we are computer scientists, we want to do this as efficiently as possible.

Navigating Trees

Often, we want to visit each vertex in a tree, or find a specific vertex. Because we are computer scientists, we want to do this as efficiently as possible.

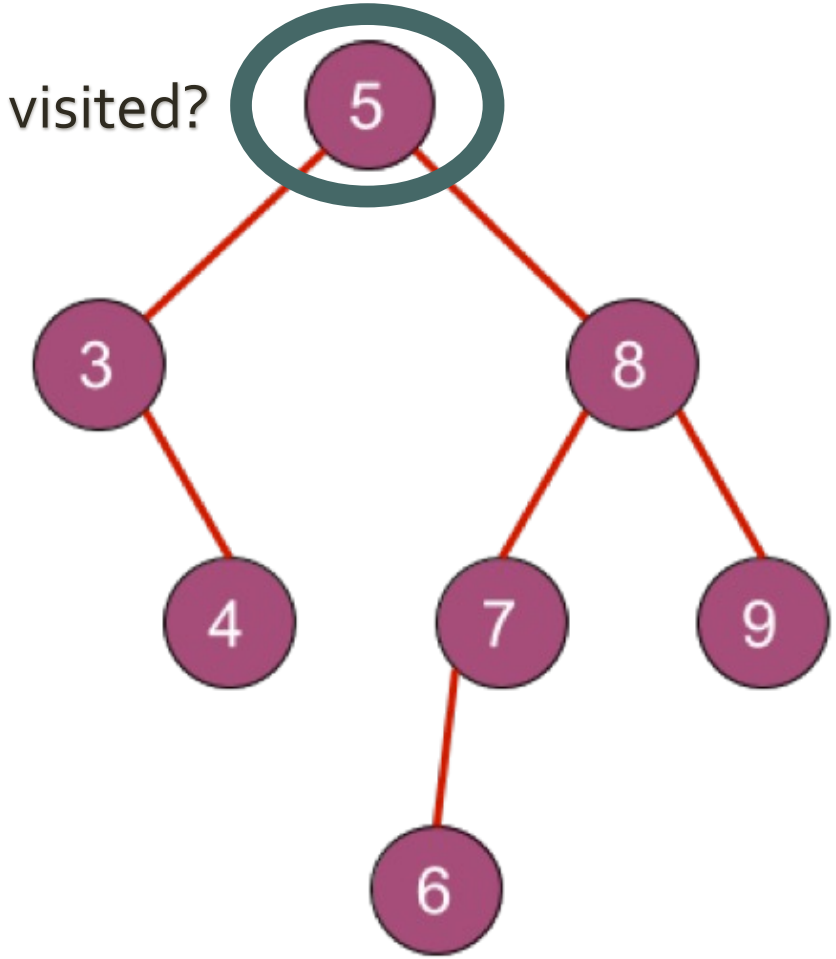
There are two common algorithms for obtaining these goals: breadth first search (BFS), and depth first search (DFS).

BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.

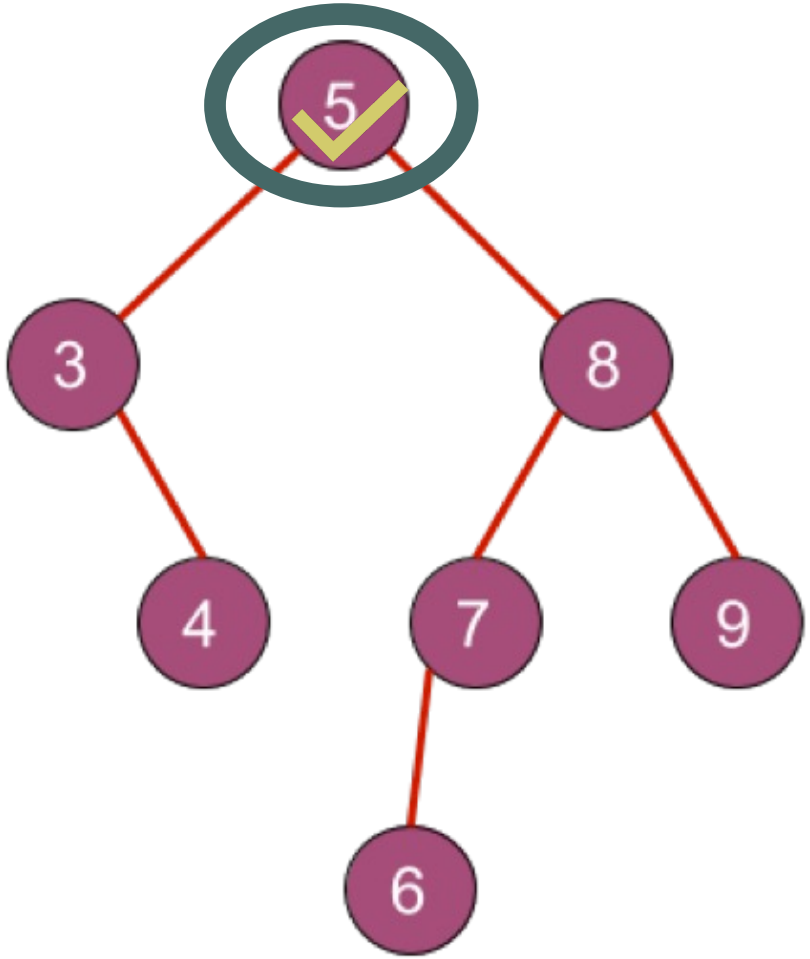
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



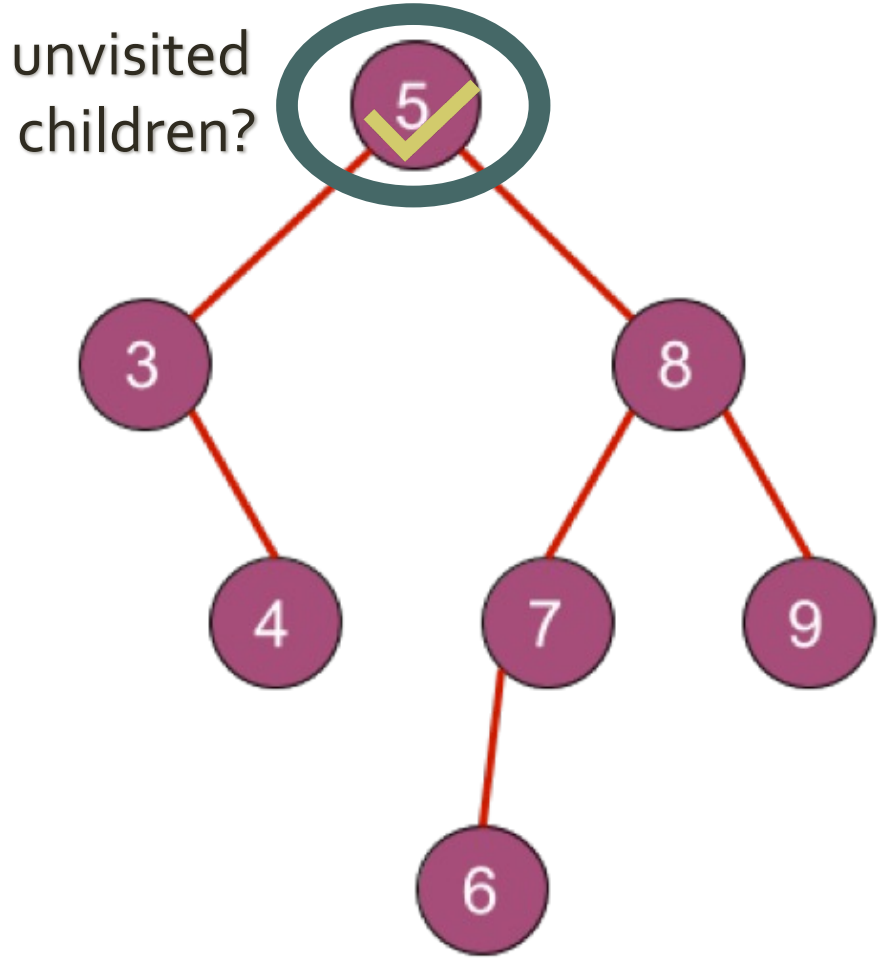
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



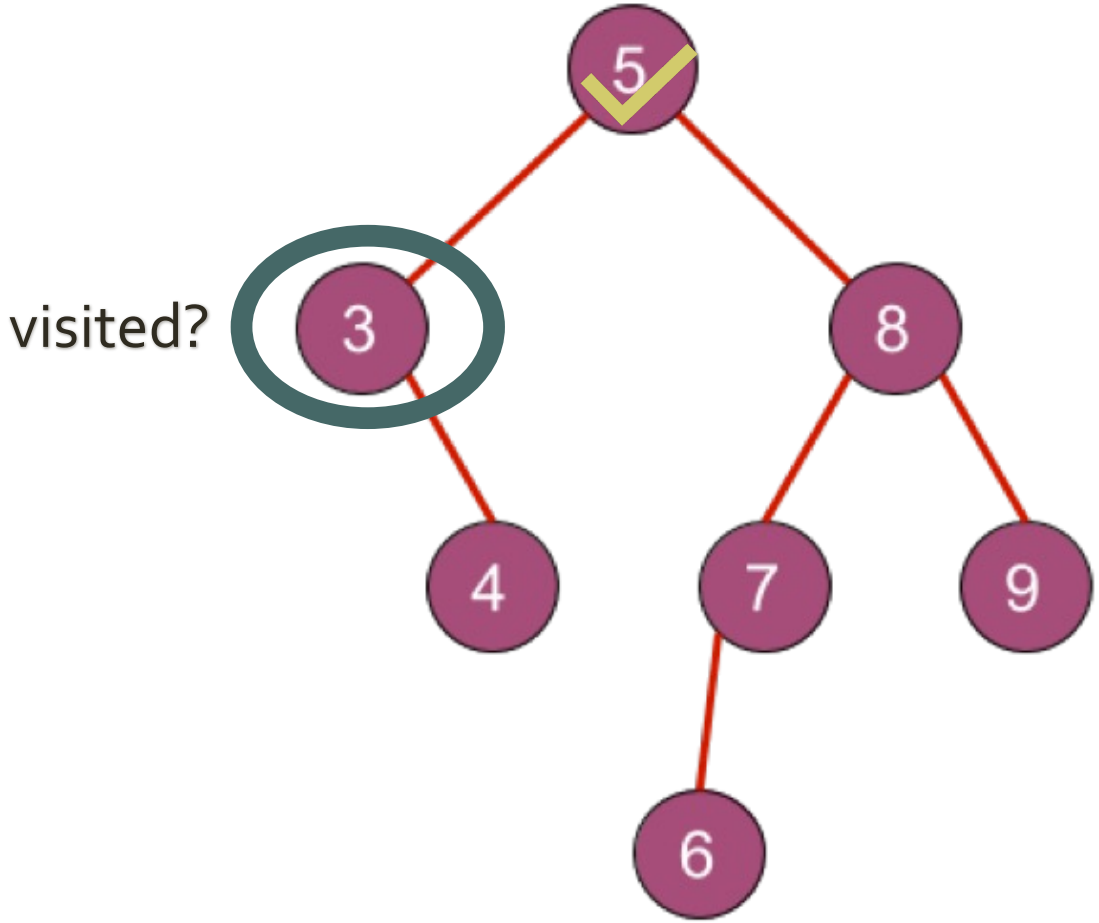
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



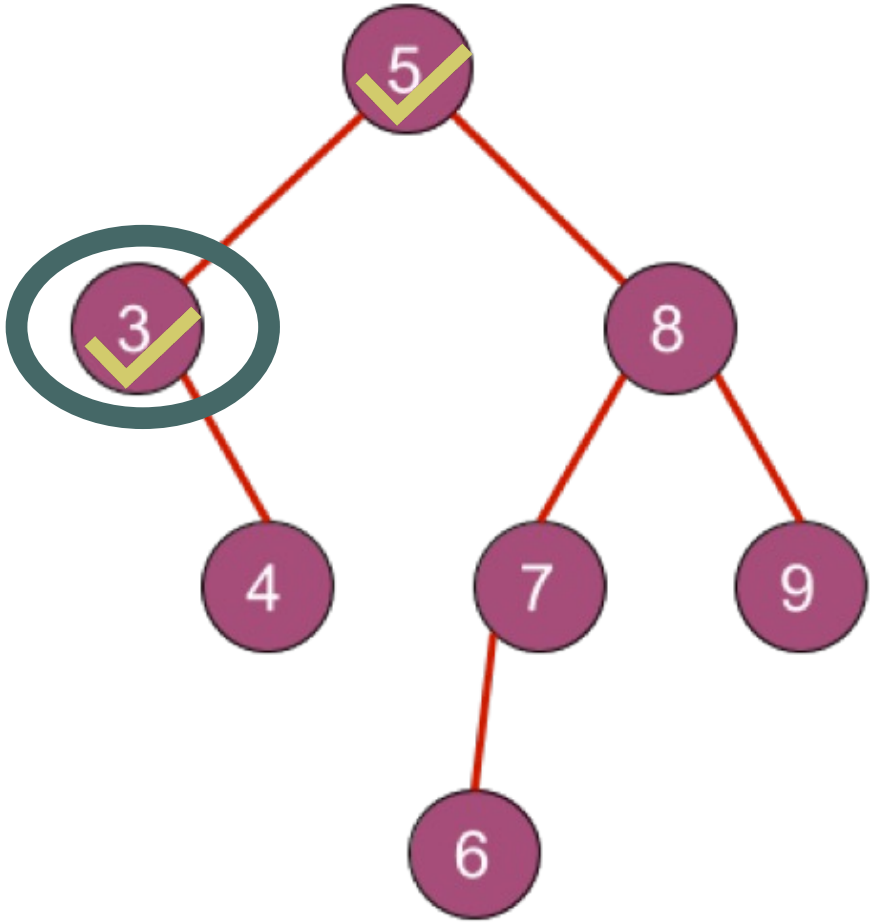
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



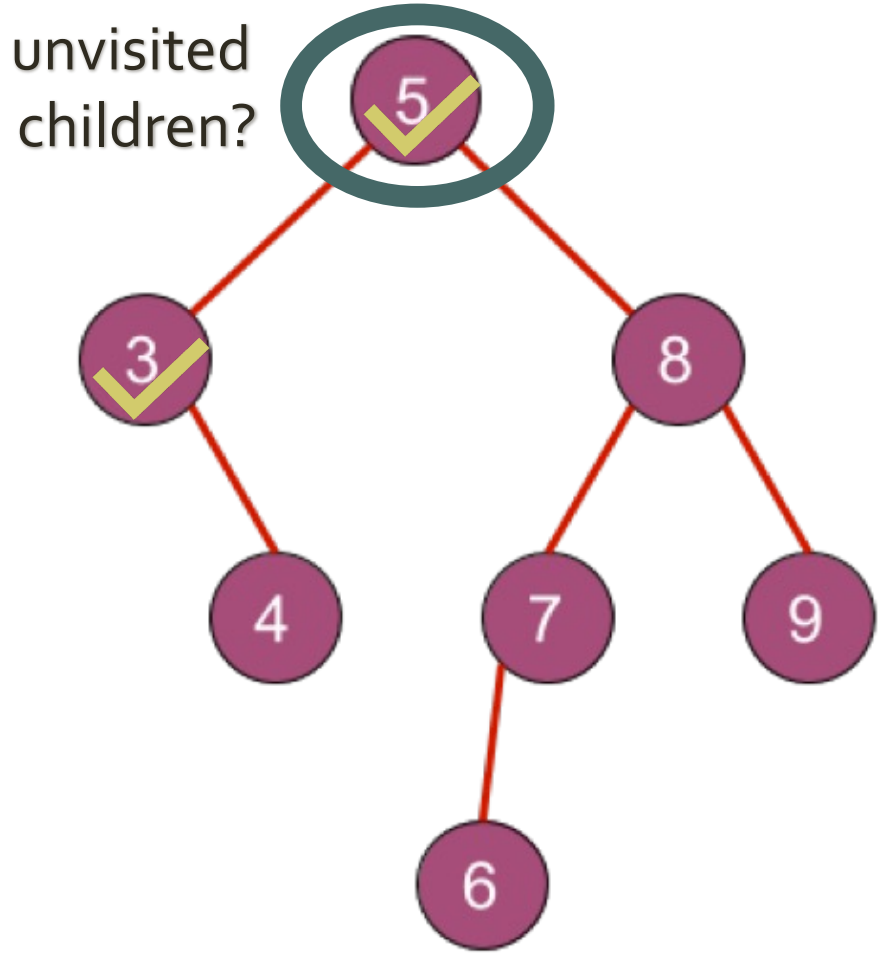
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



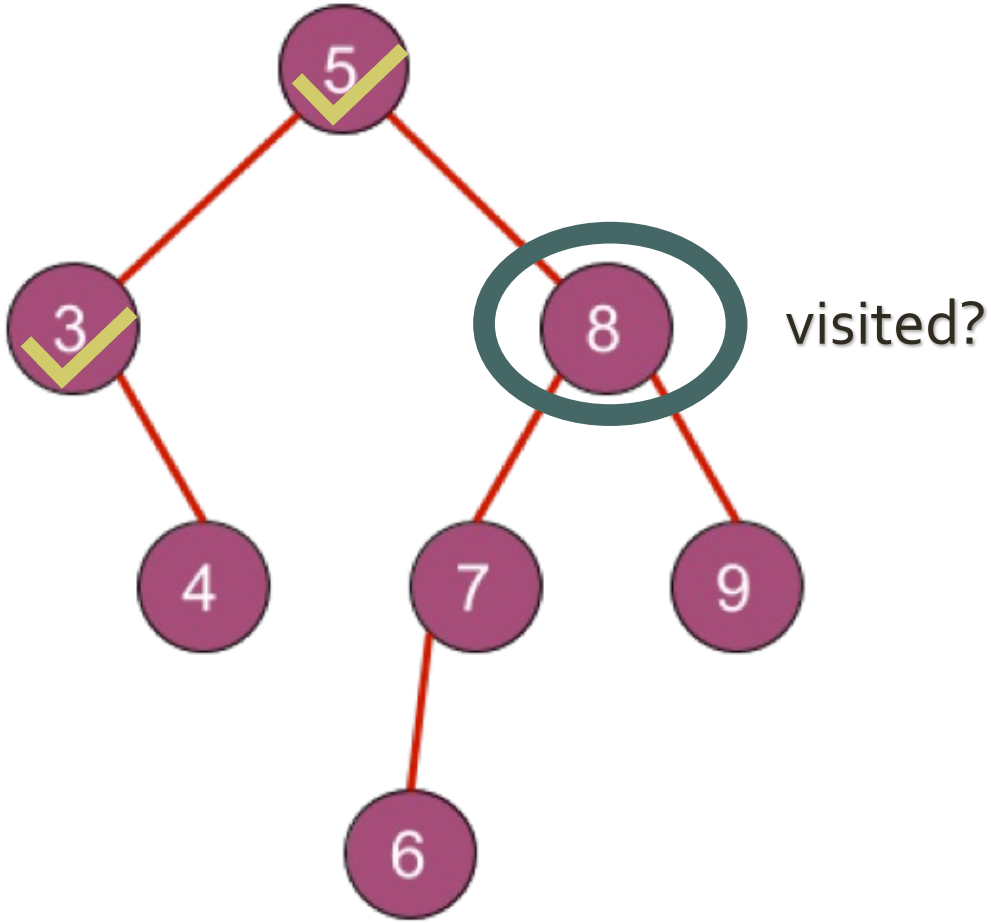
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



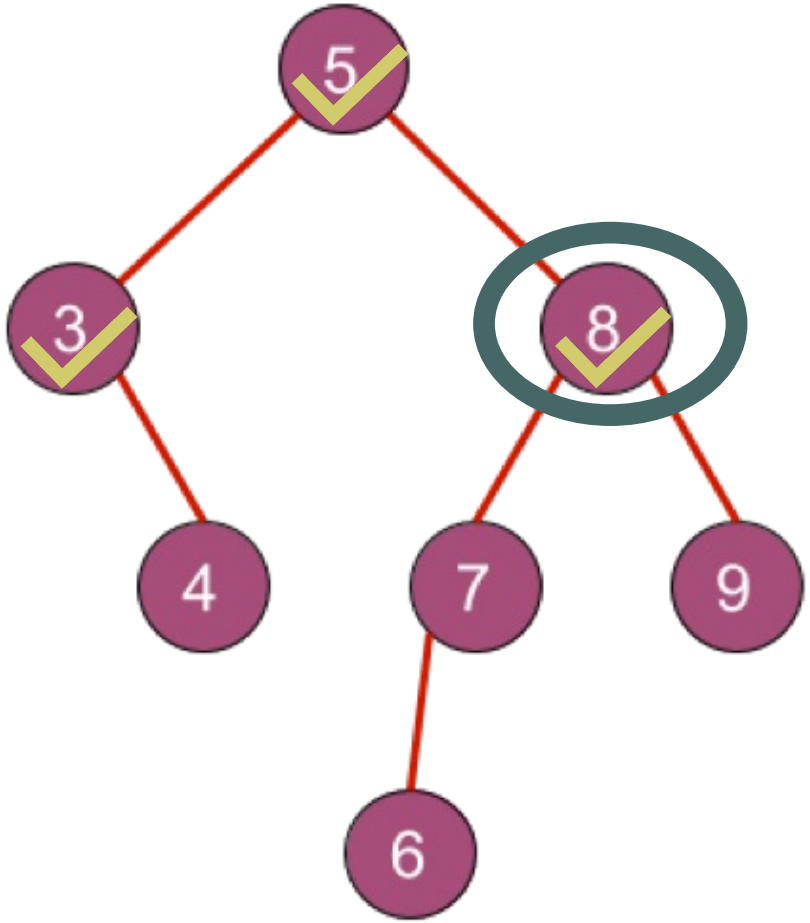
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



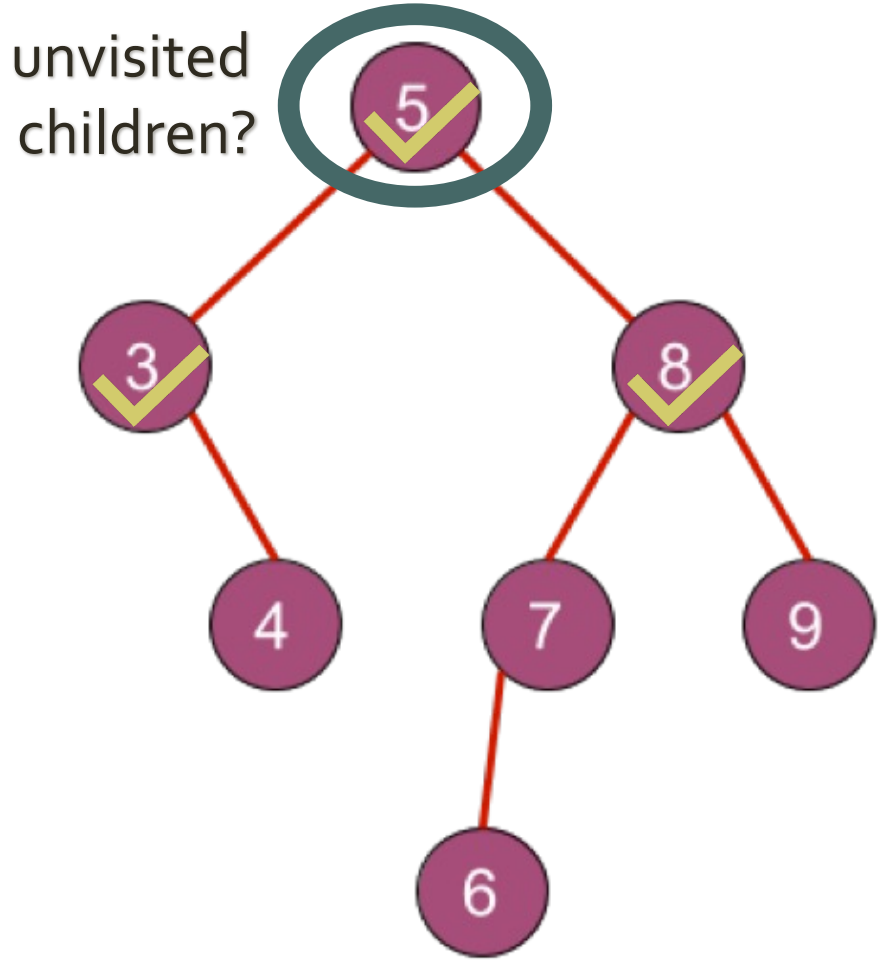
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



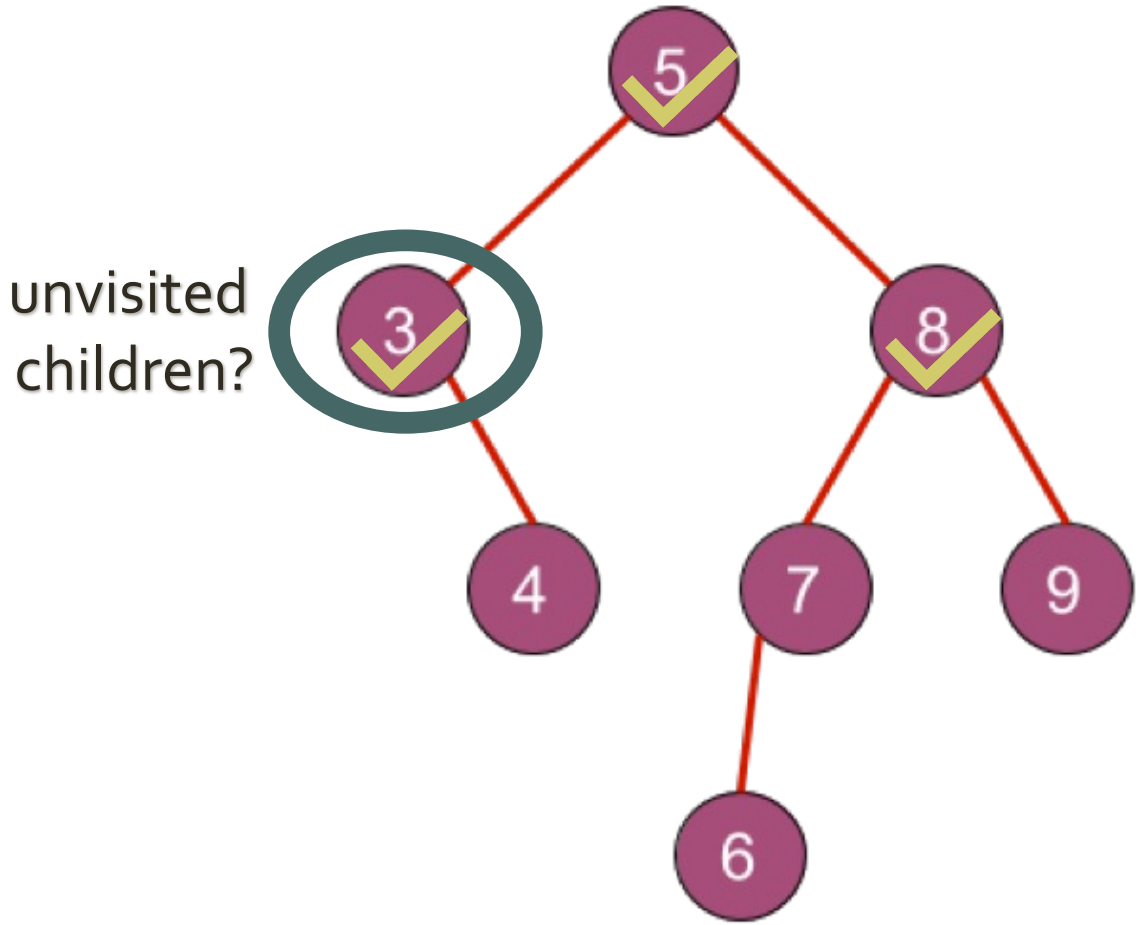
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



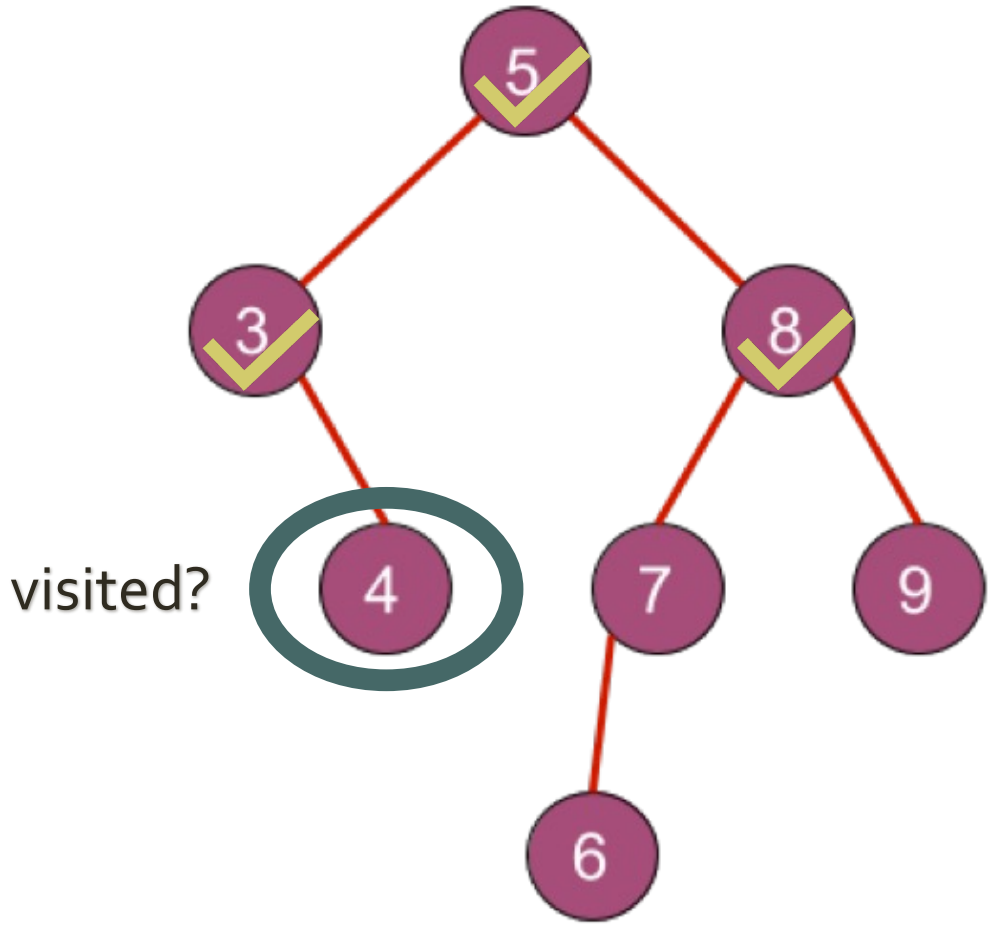
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



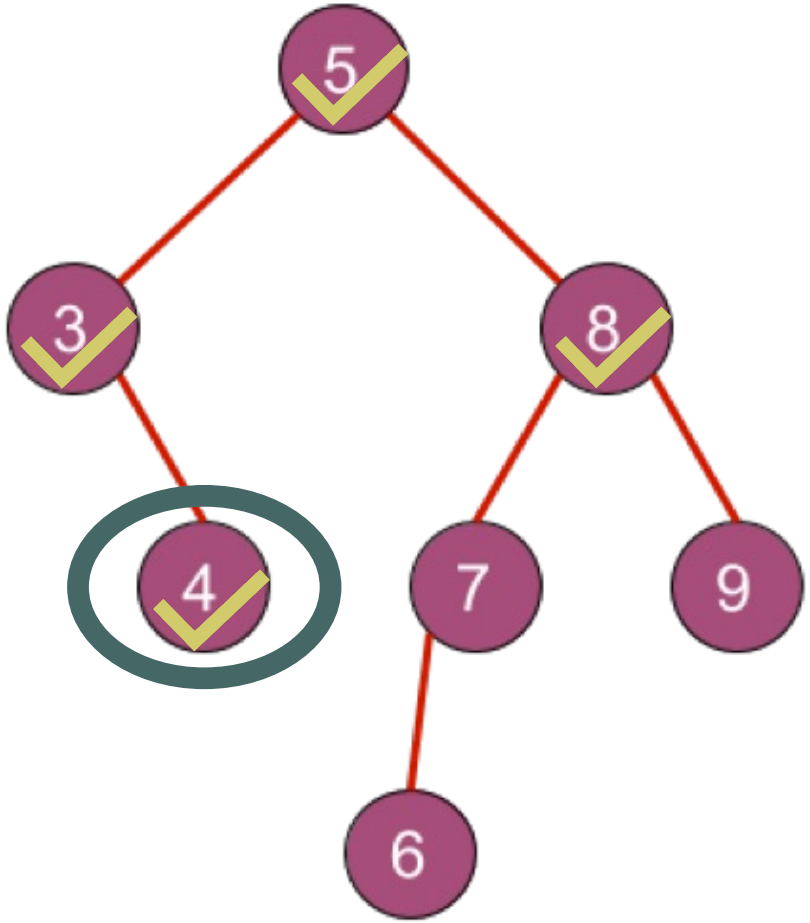
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



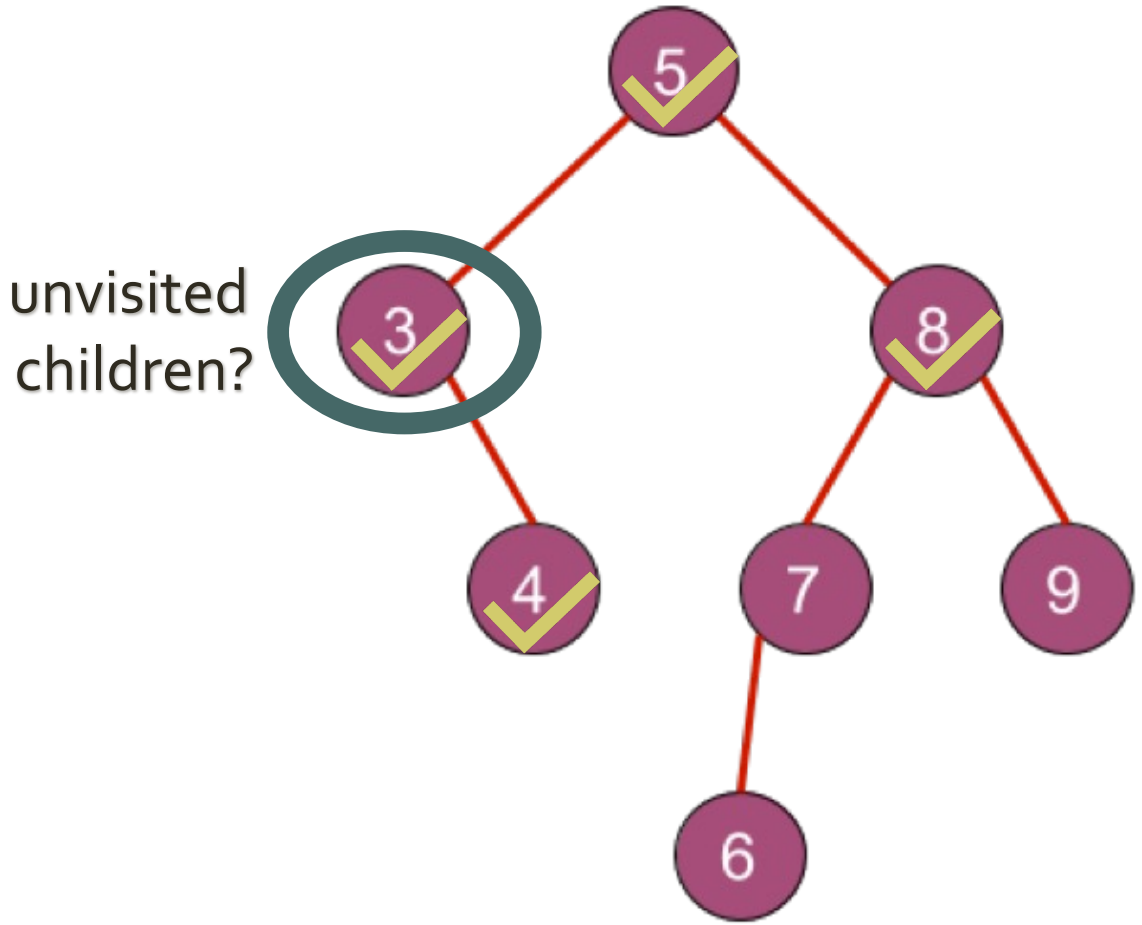
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



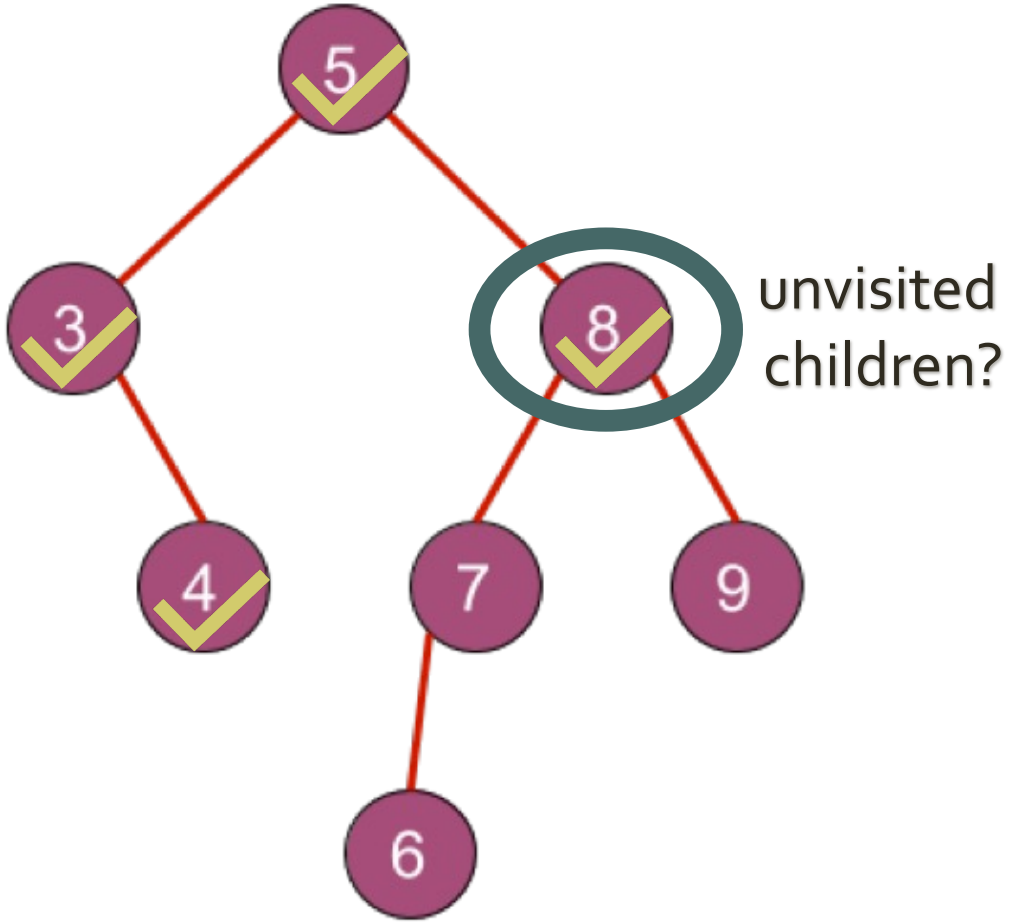
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



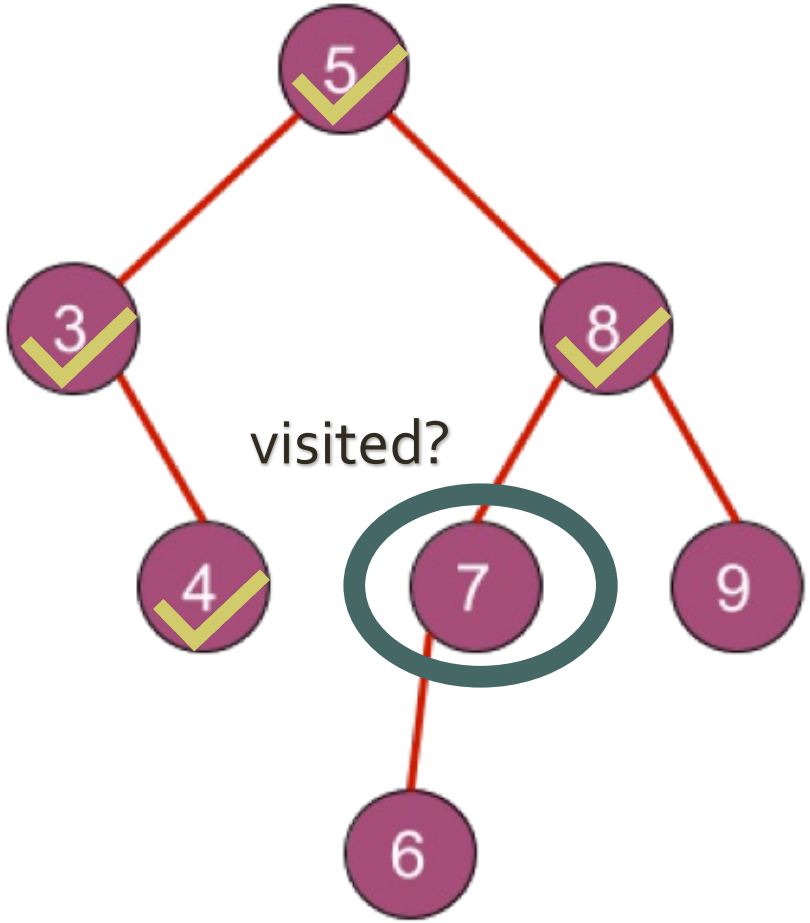
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



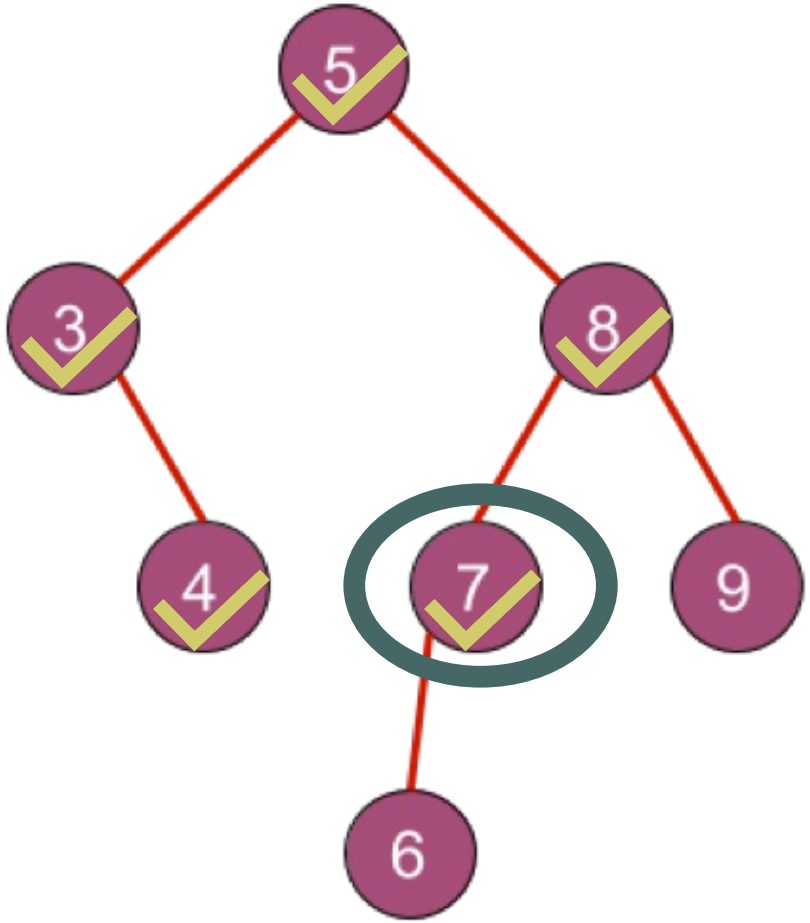
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



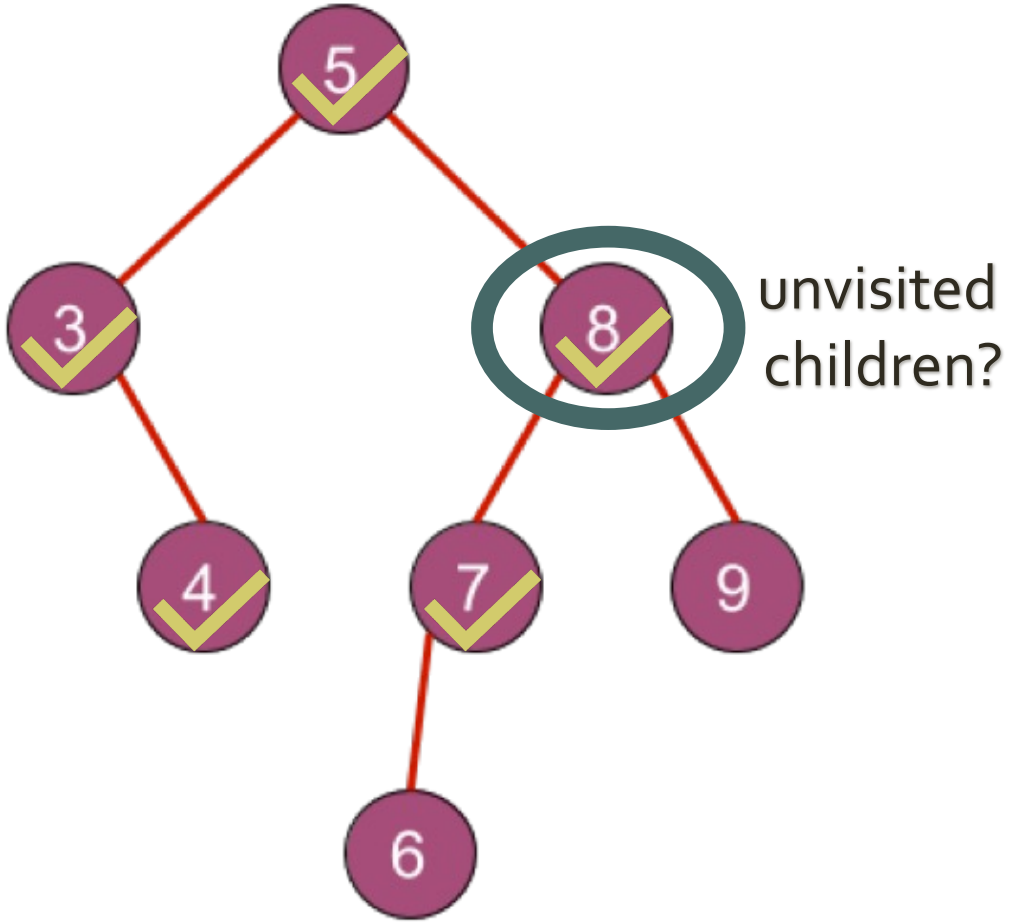
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



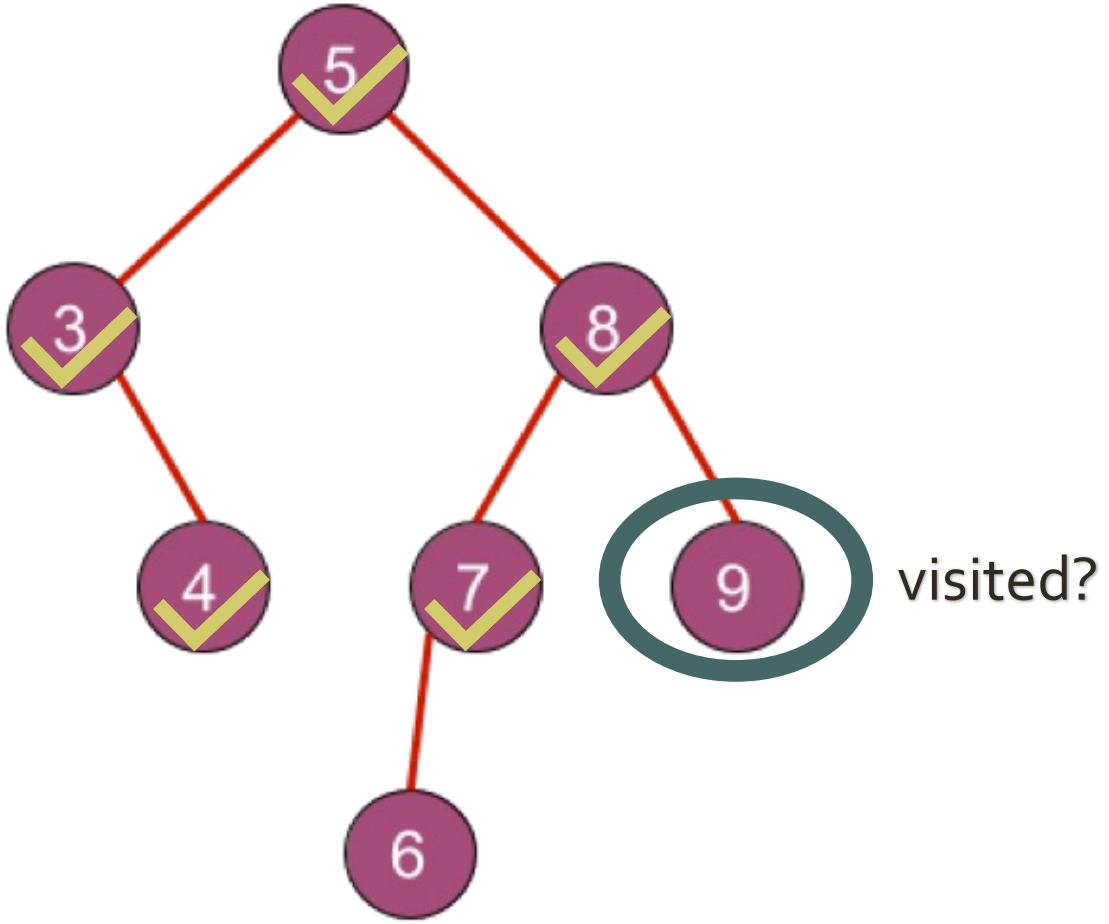
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



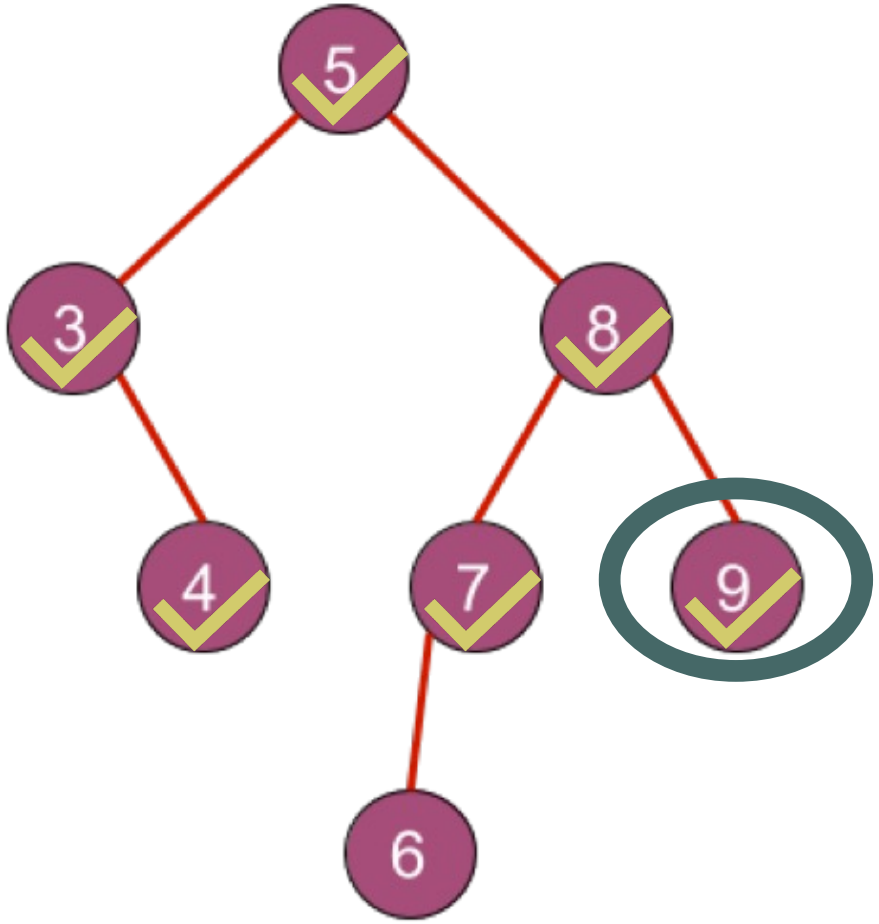
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



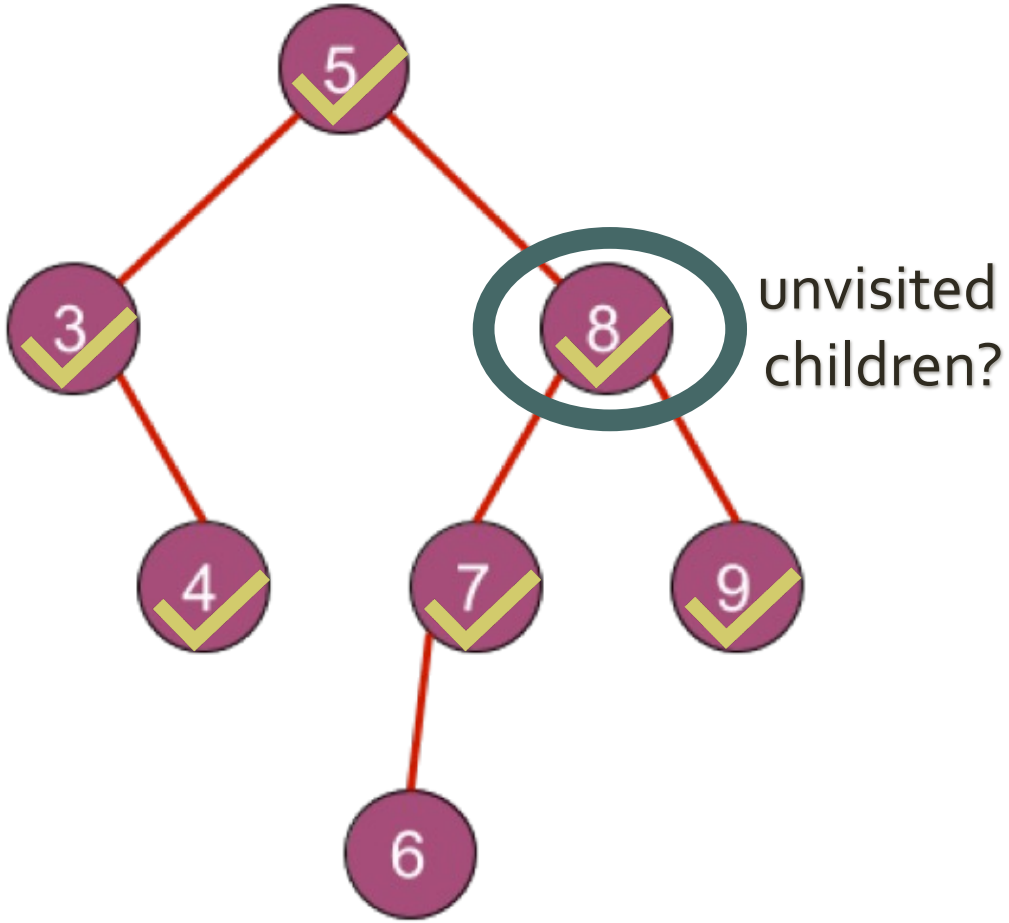
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



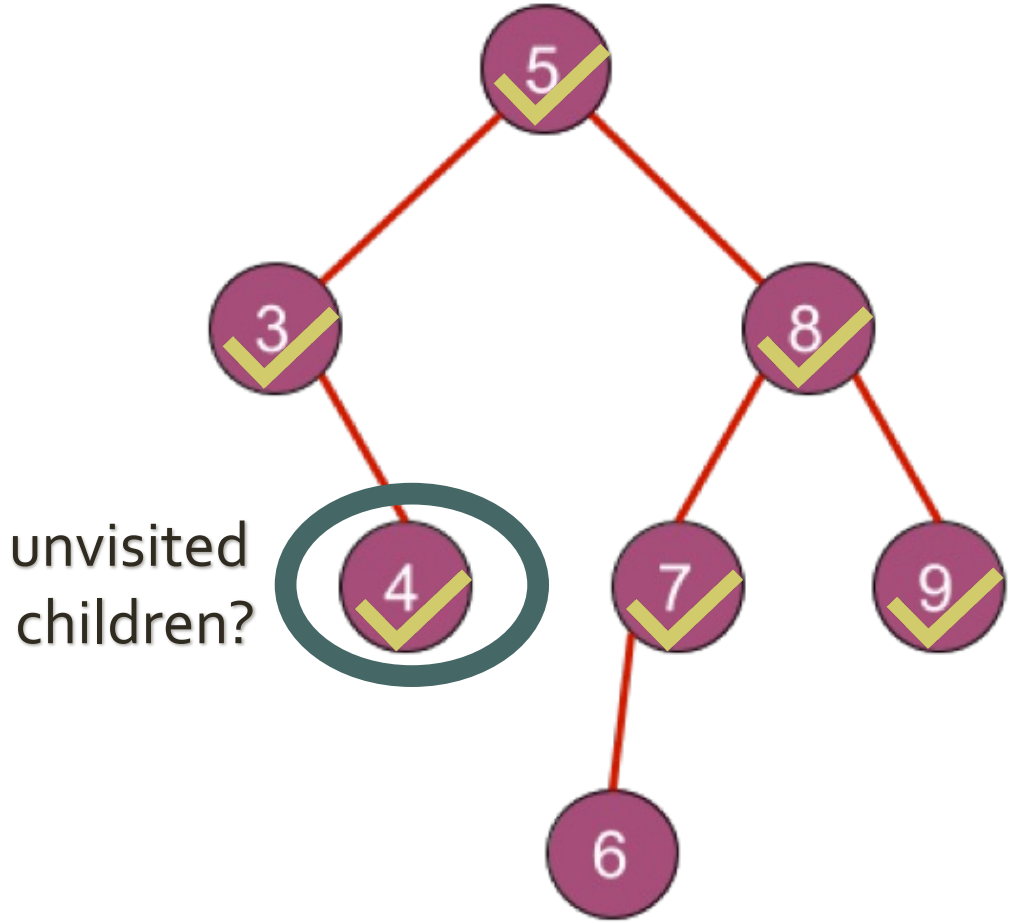
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



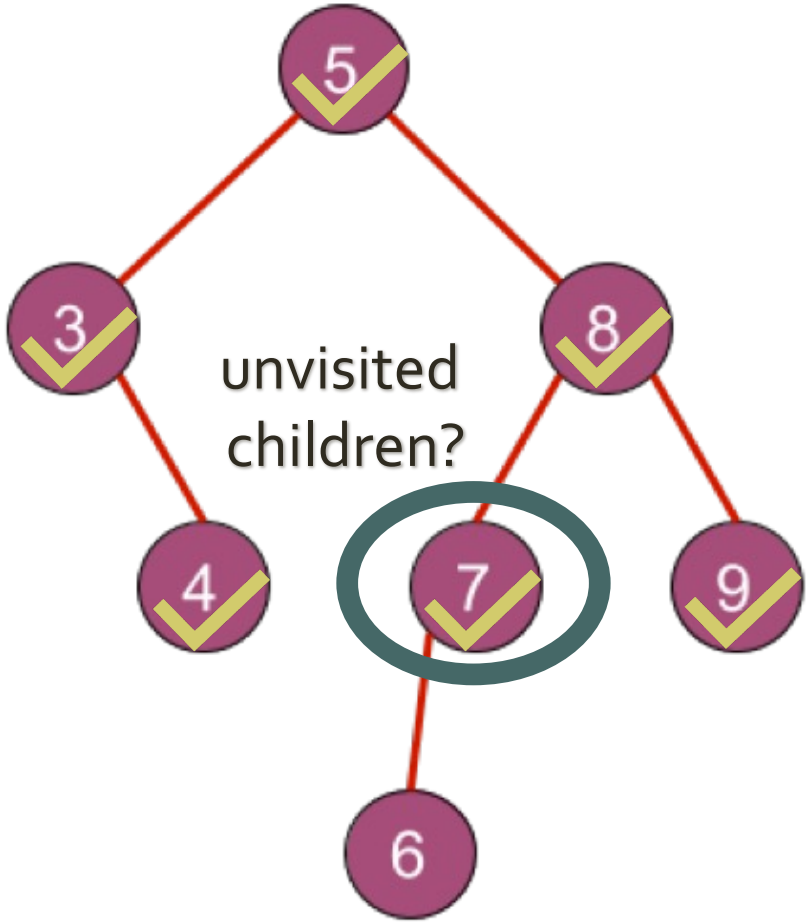
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



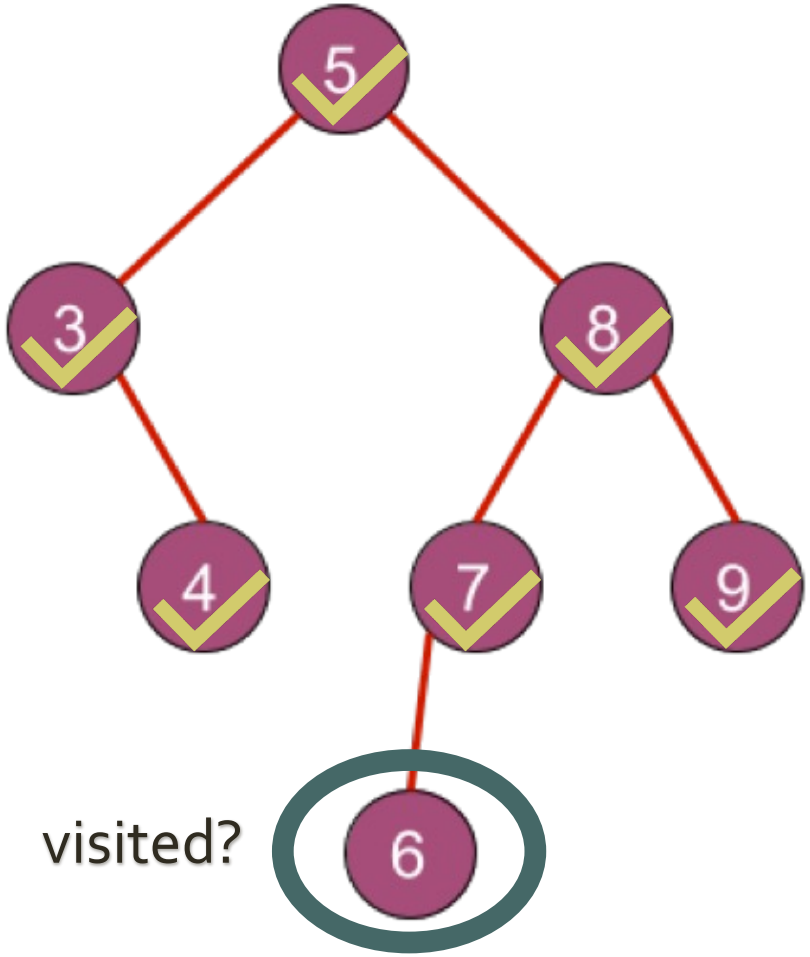
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



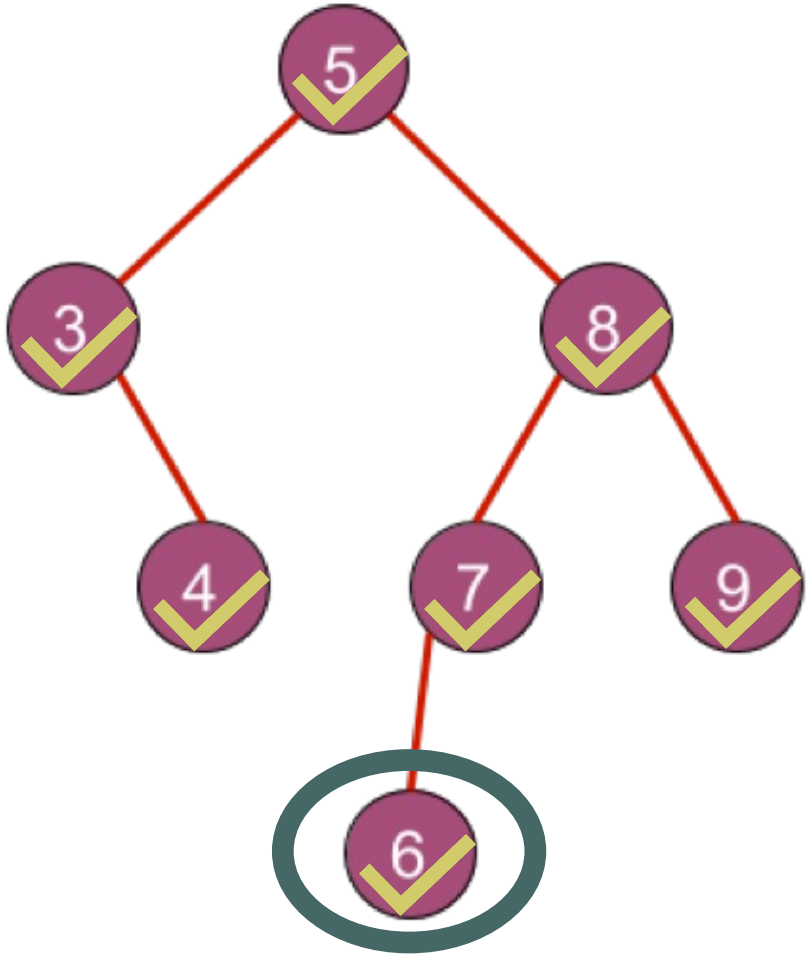
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



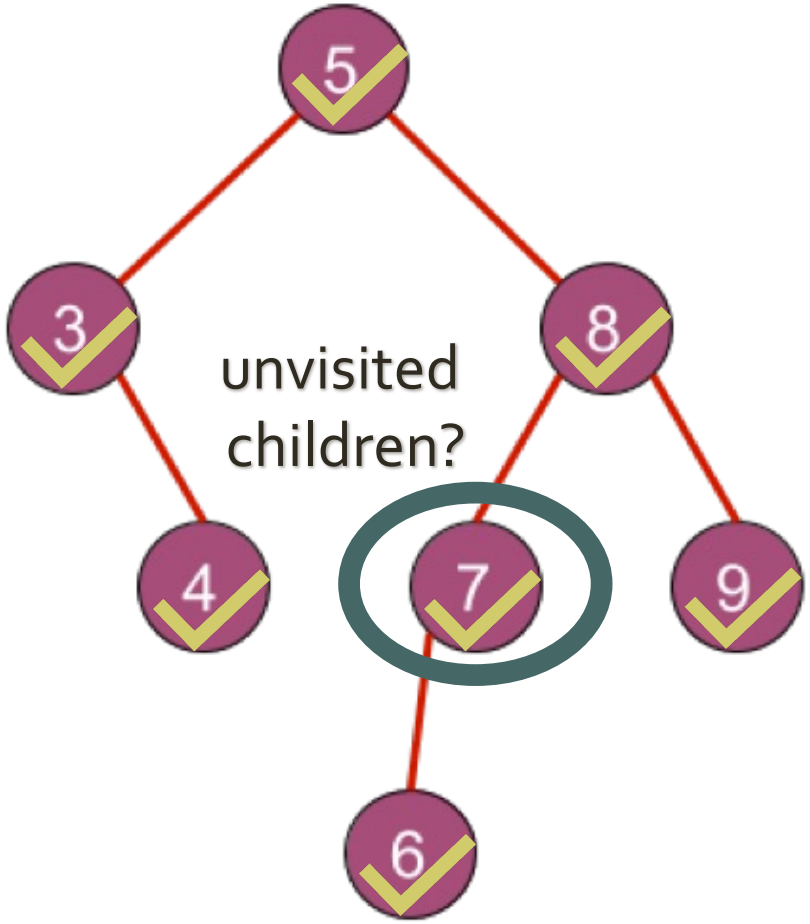
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



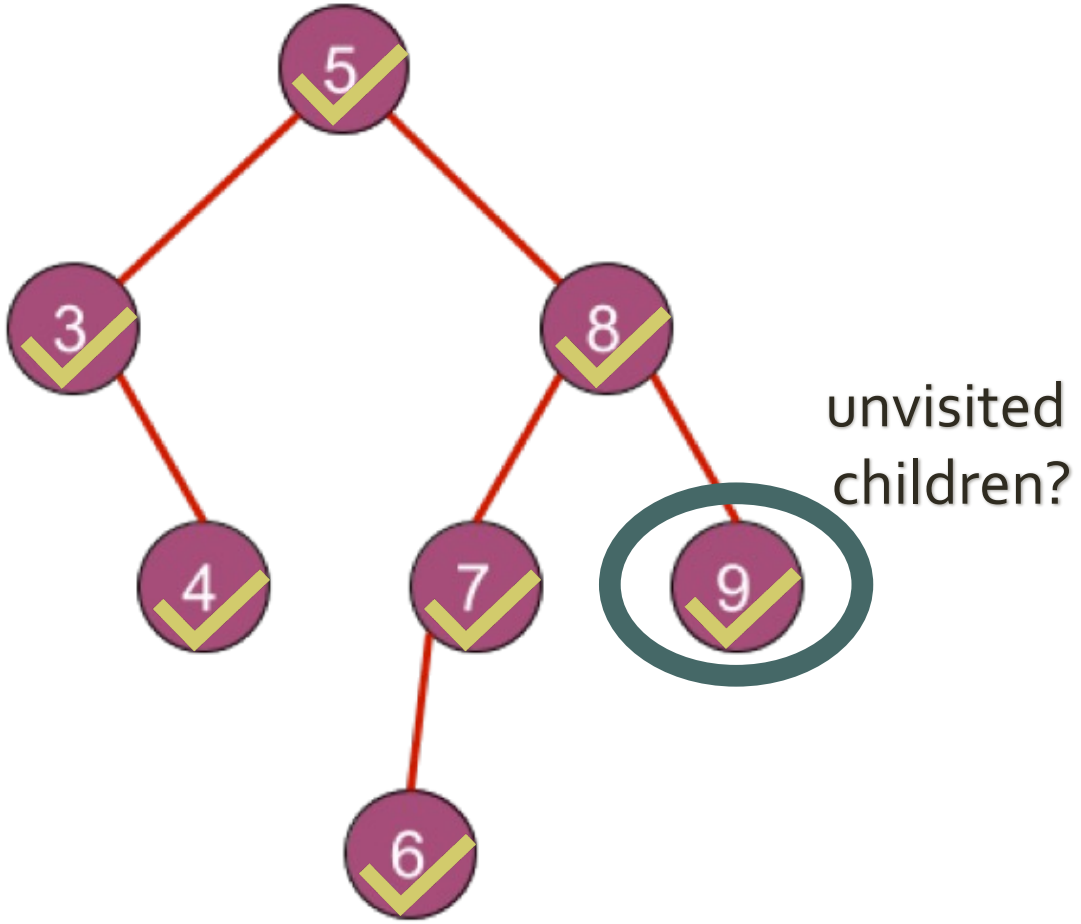
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



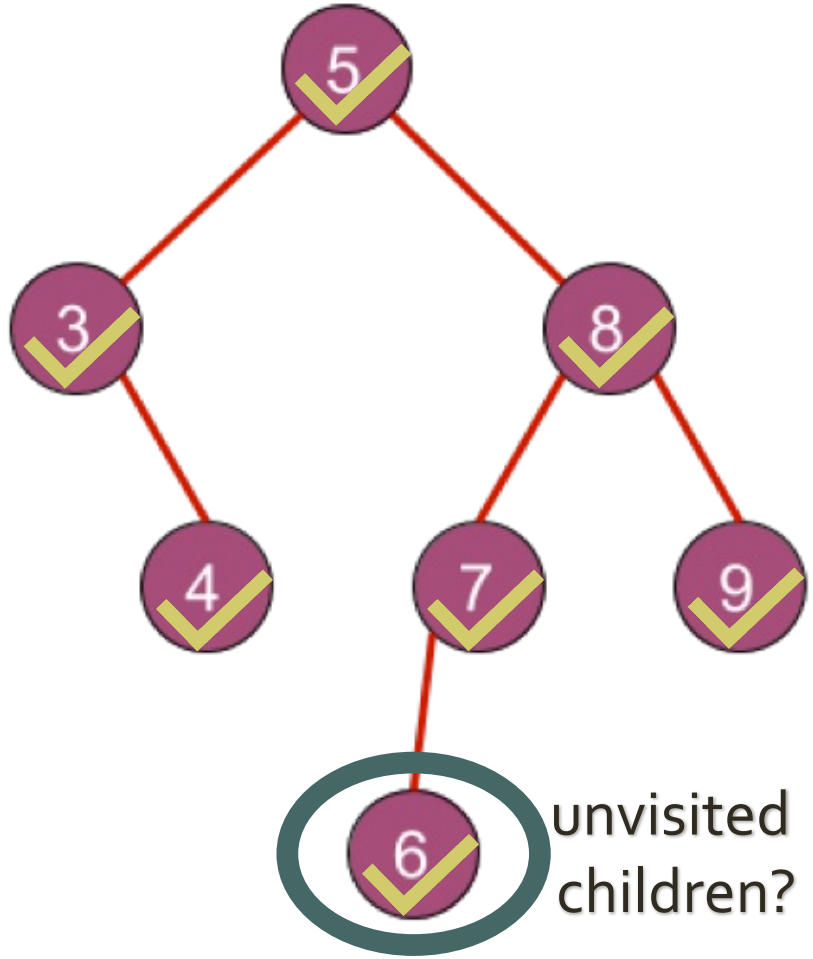
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



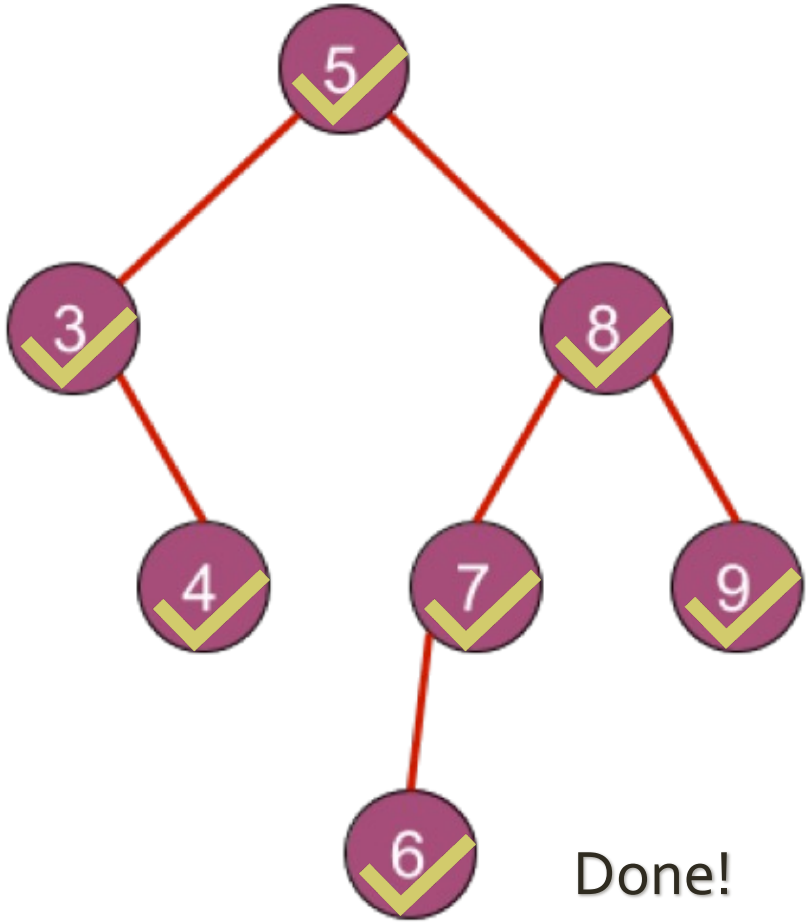
BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.



BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.

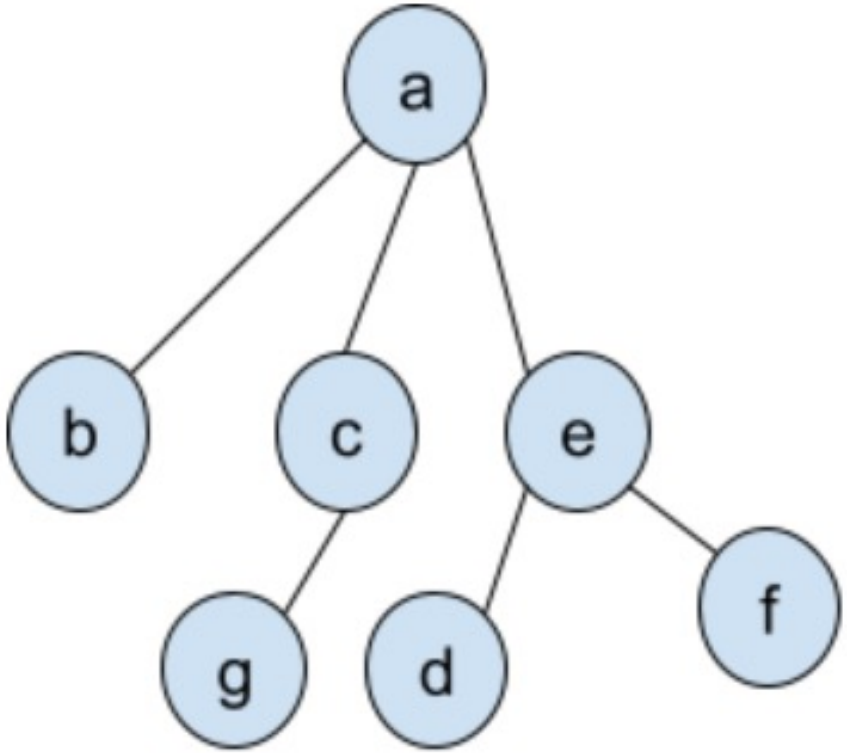
Algorithm:

```
BFS(G, v)
    create a queue Q
    mark v as visited and add to Q
    while Q is non-empty
        remove the head u of Q
        mark and enqueue all unvisited neighbors of u
```

BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.

Draw this graph on the board and do a BFS. Count how many steps the BFS takes.



BFS

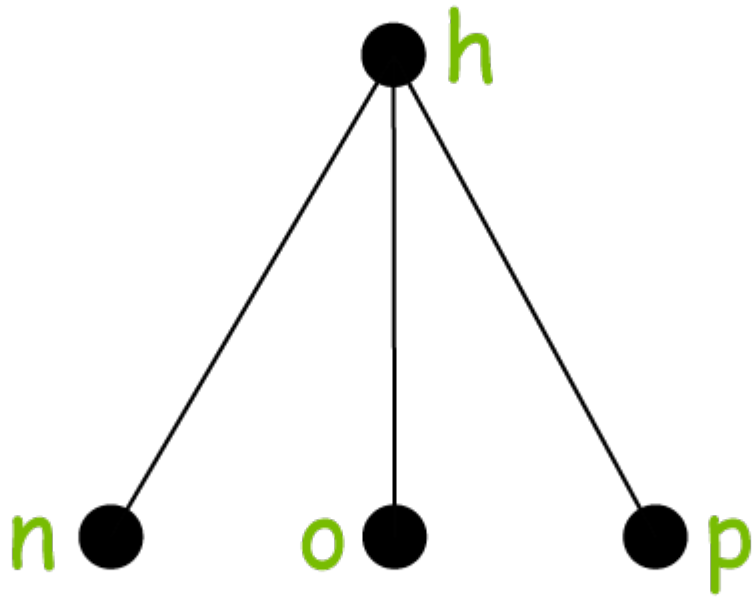
In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.

Draw this graph on the board and do a BFS. Count how many steps the BFS takes.

BFS

In a ***breadth first search (BFS)*** we visit all vertices in the same generation before any vertices of the next generation.

Draw this graph on the board and do a BFS. Count how many steps the BFS takes.

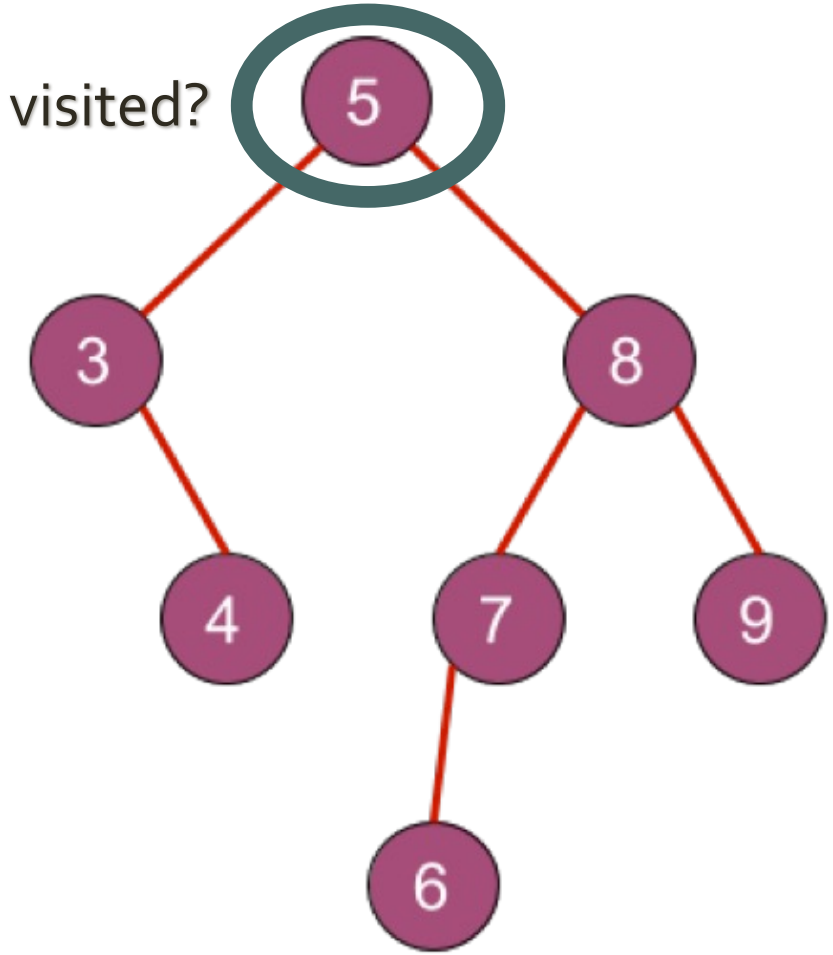


DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.

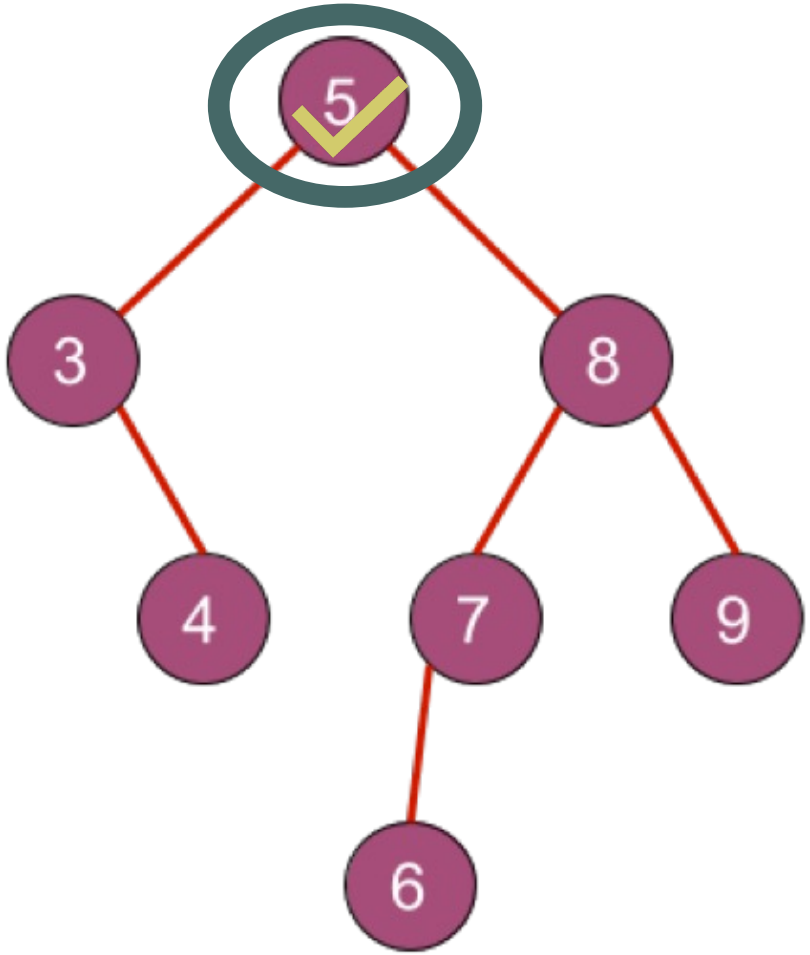
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



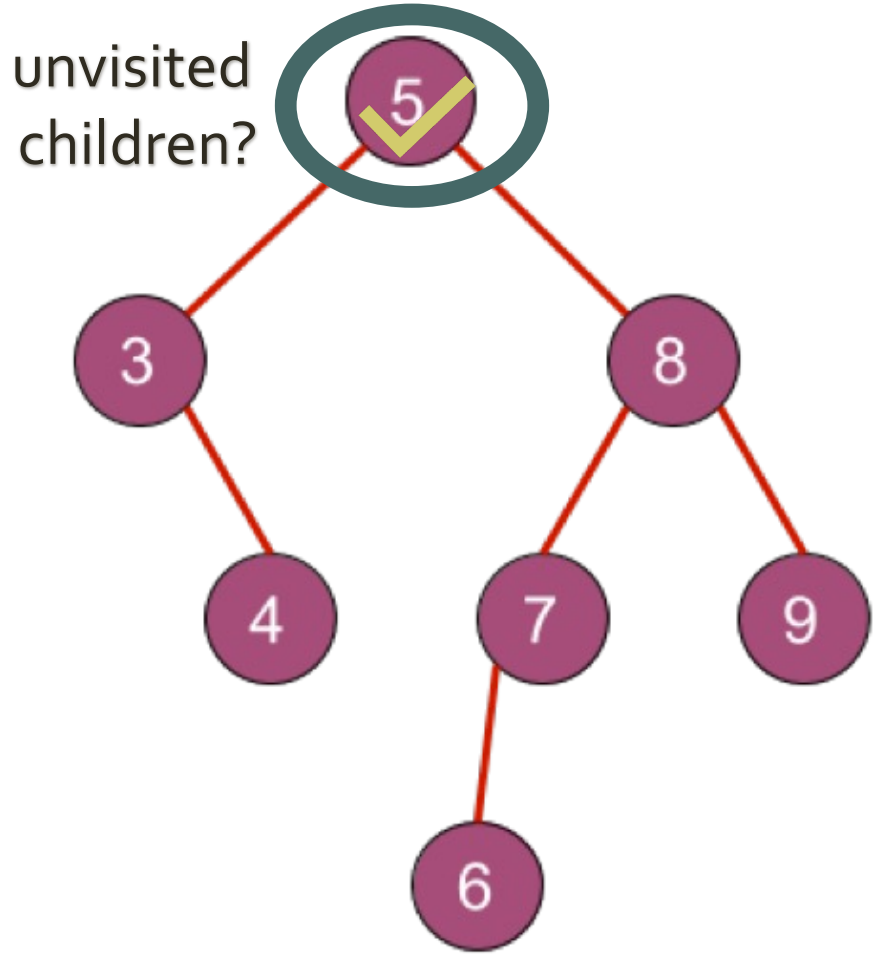
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



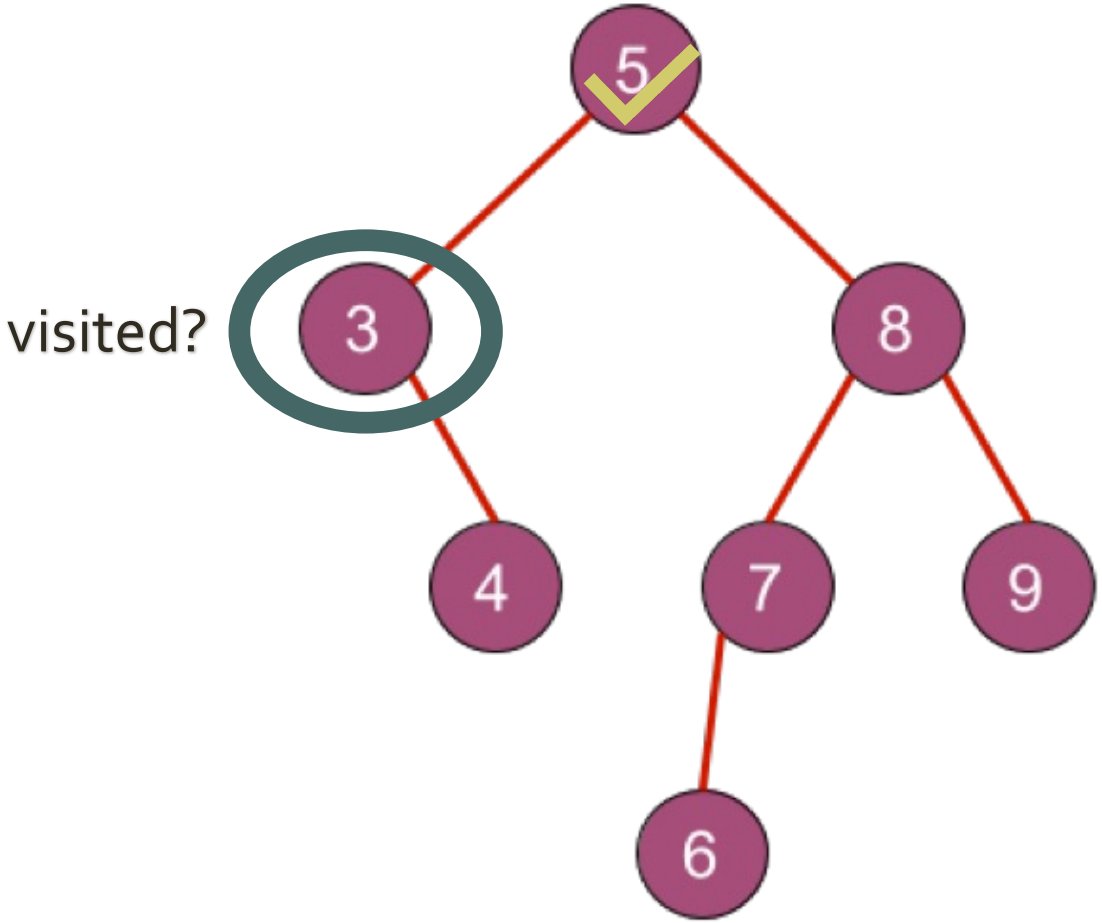
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



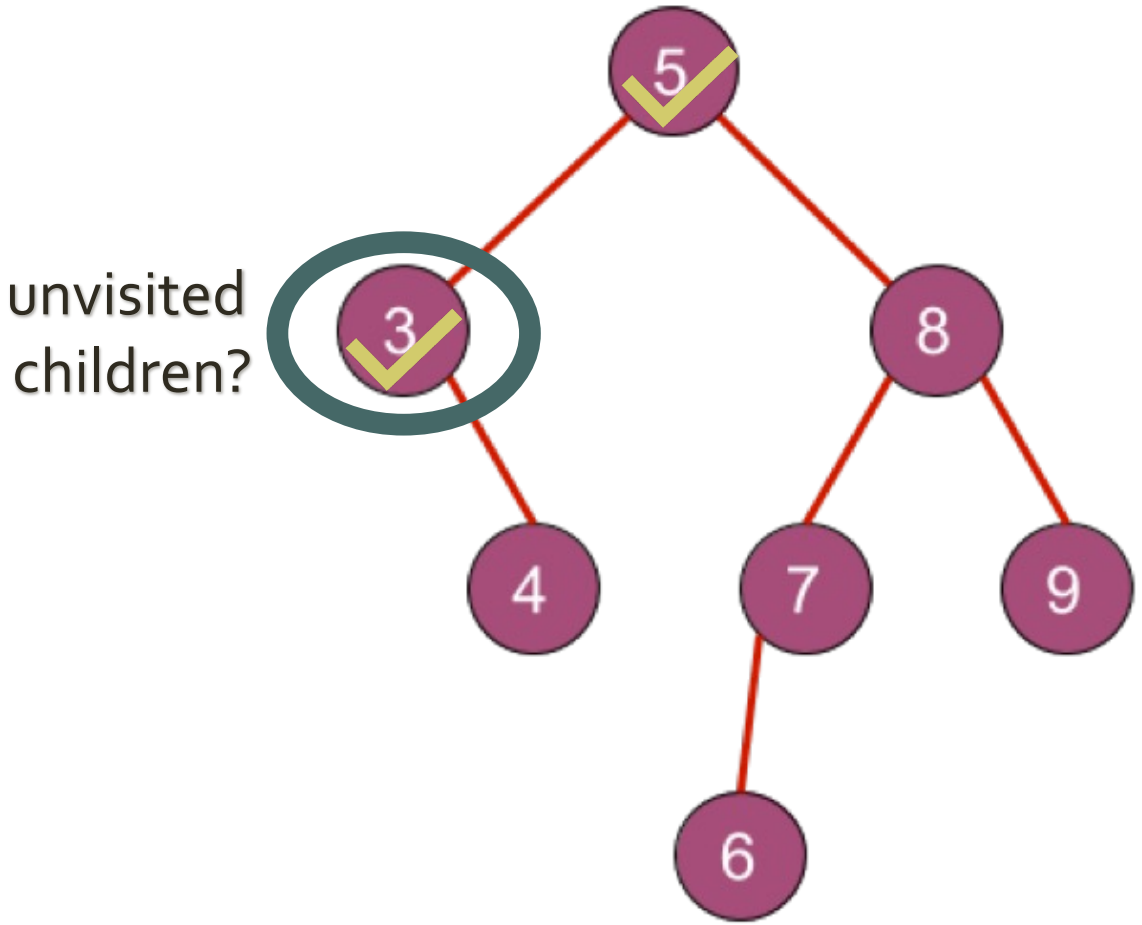
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



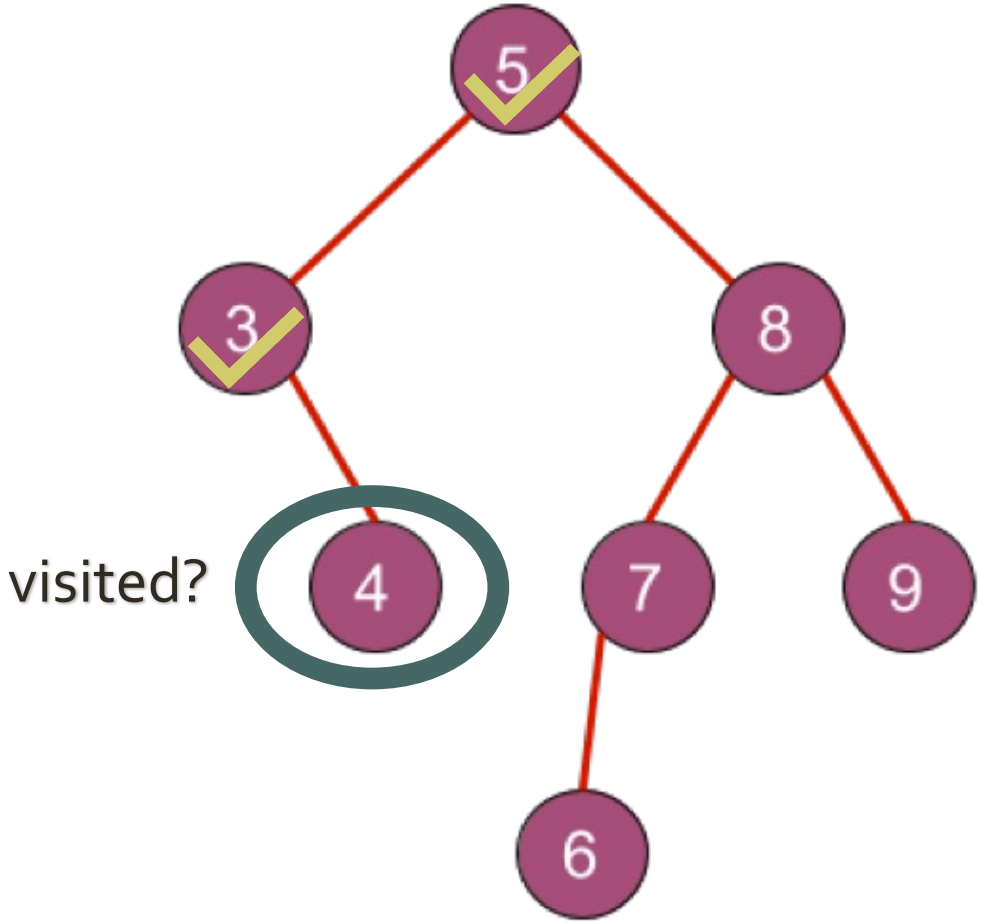
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



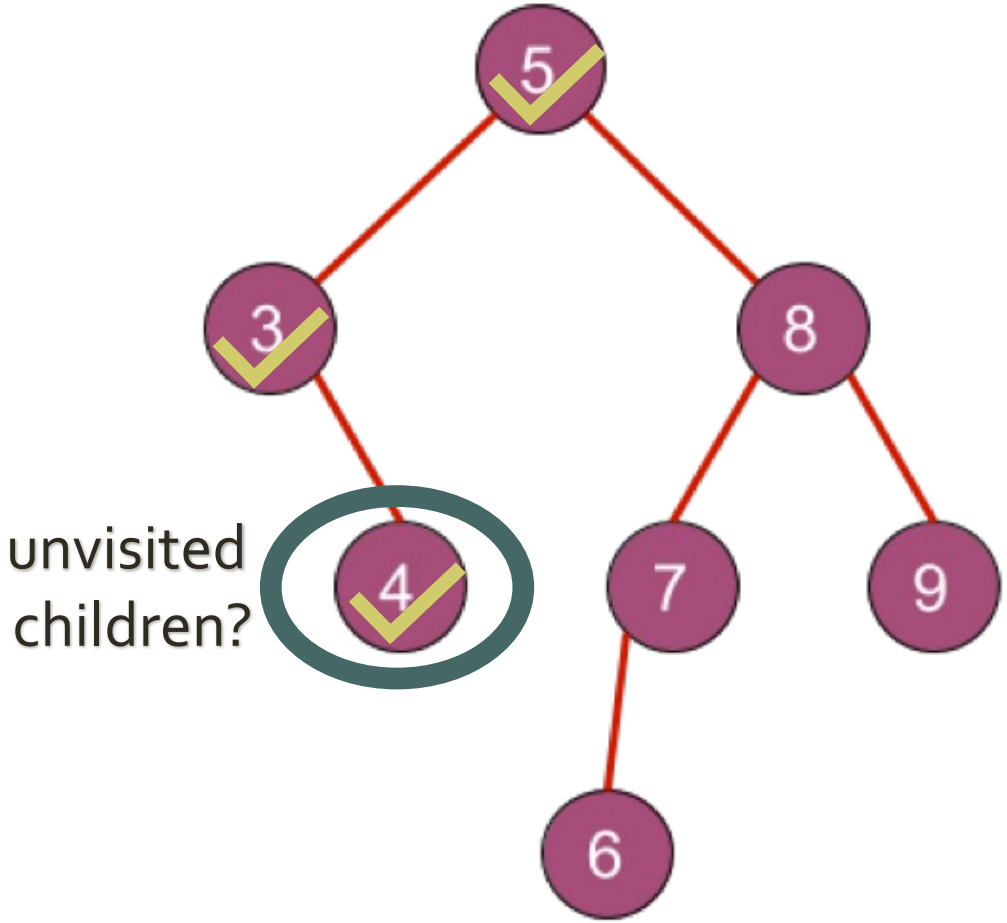
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



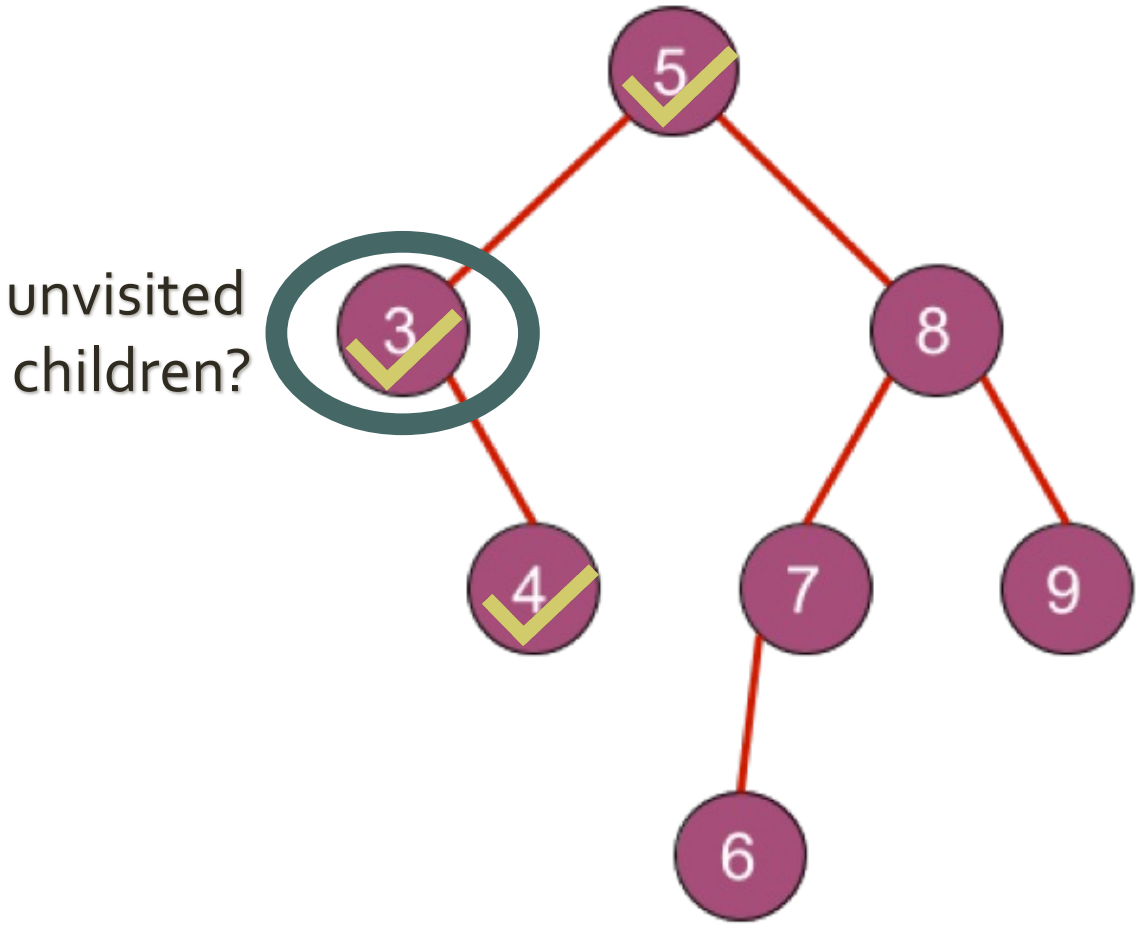
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



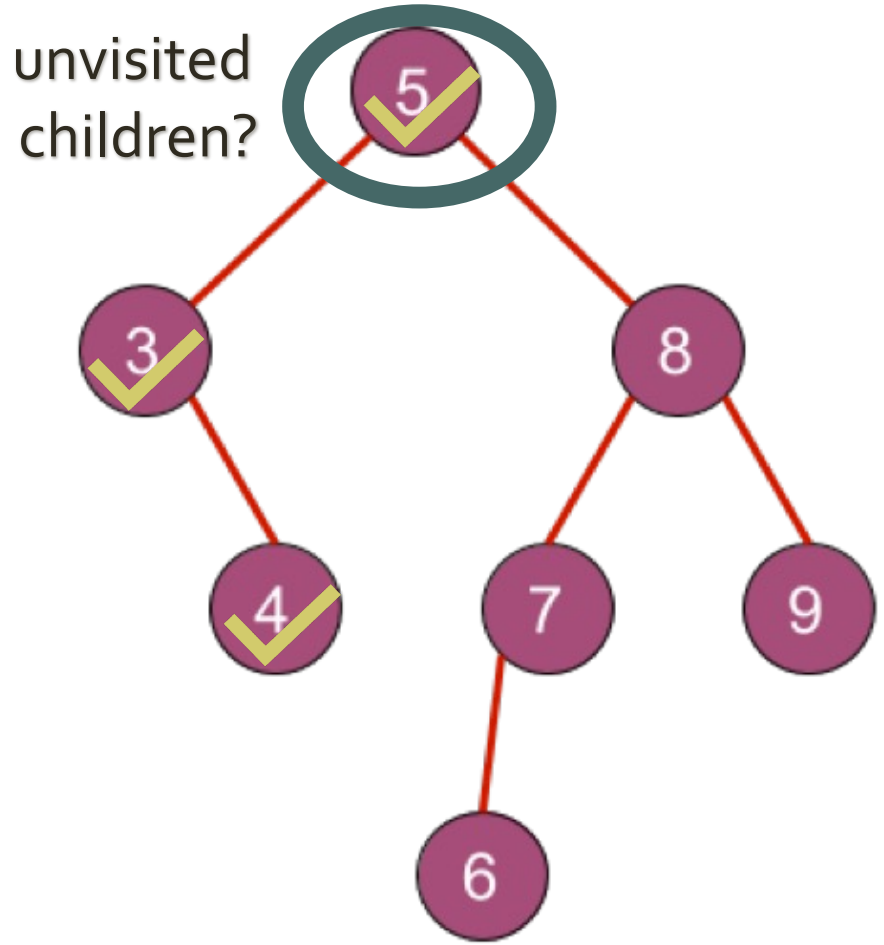
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



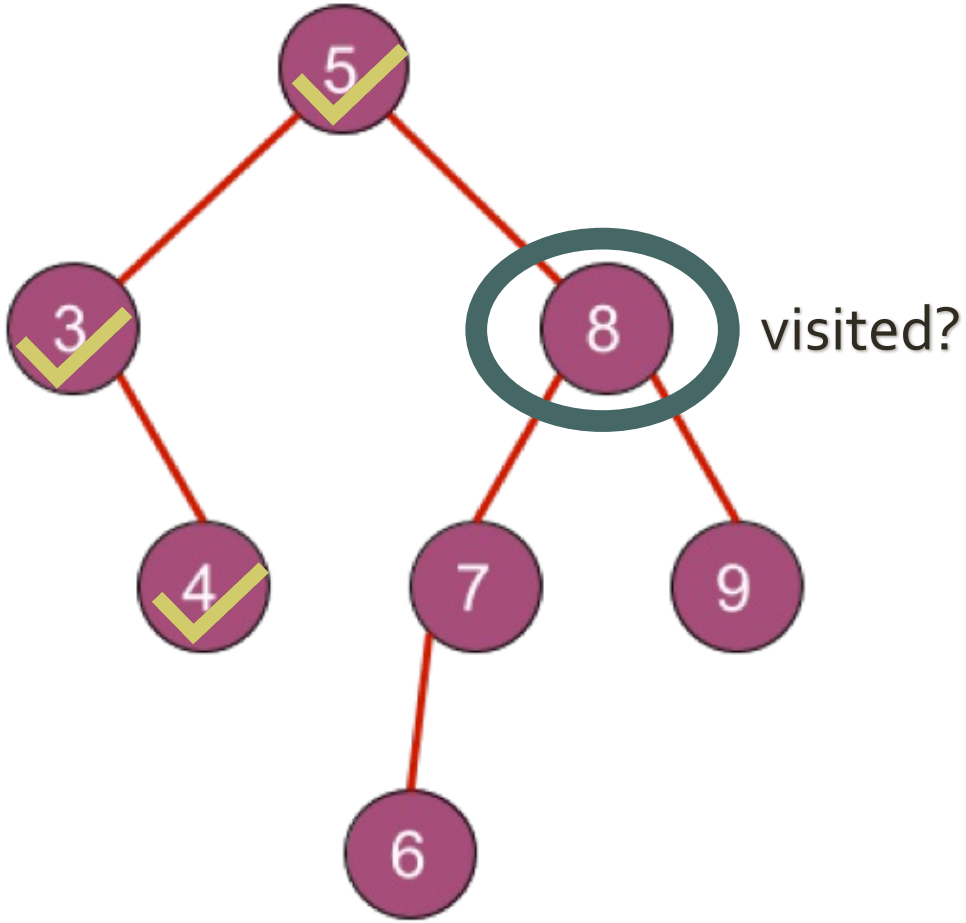
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



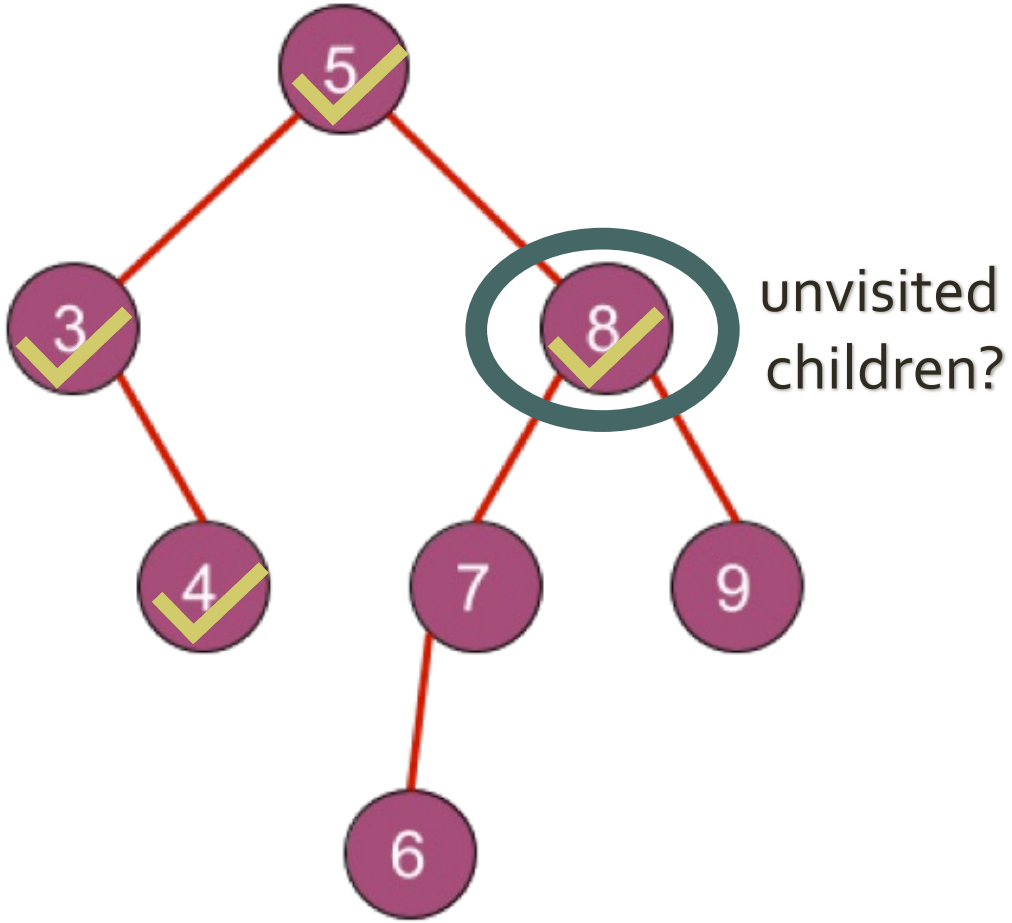
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



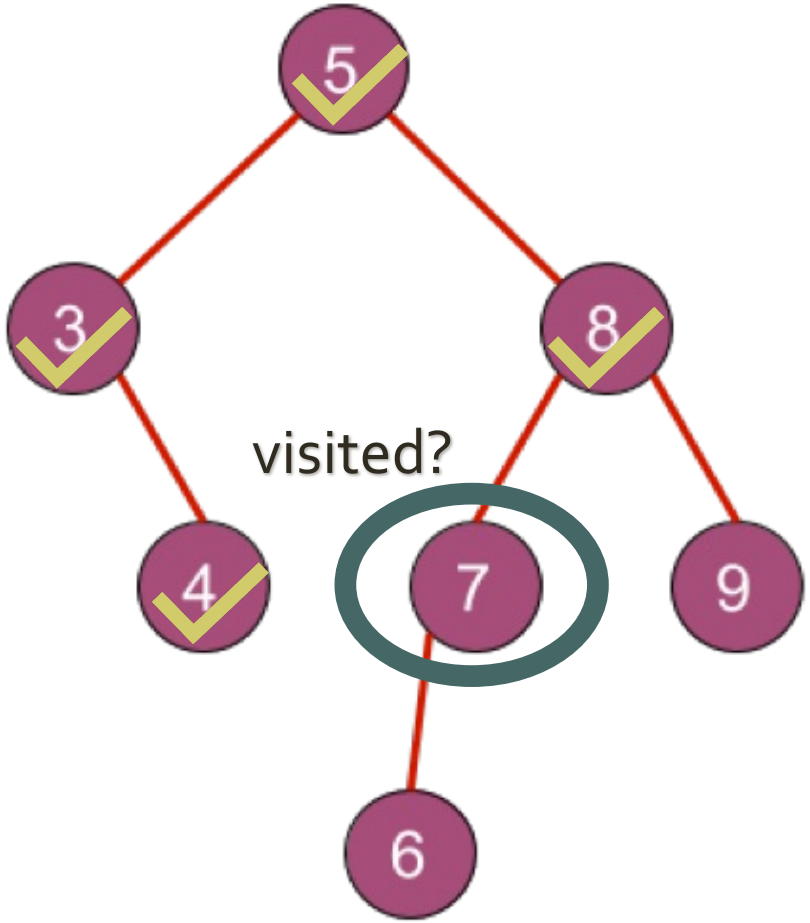
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



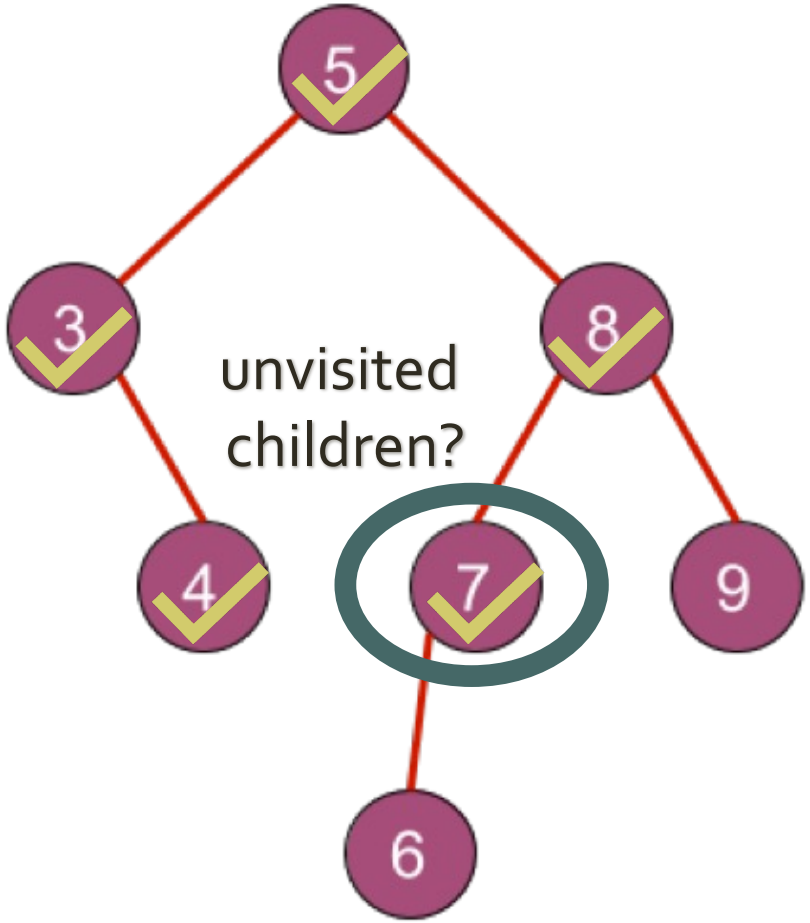
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



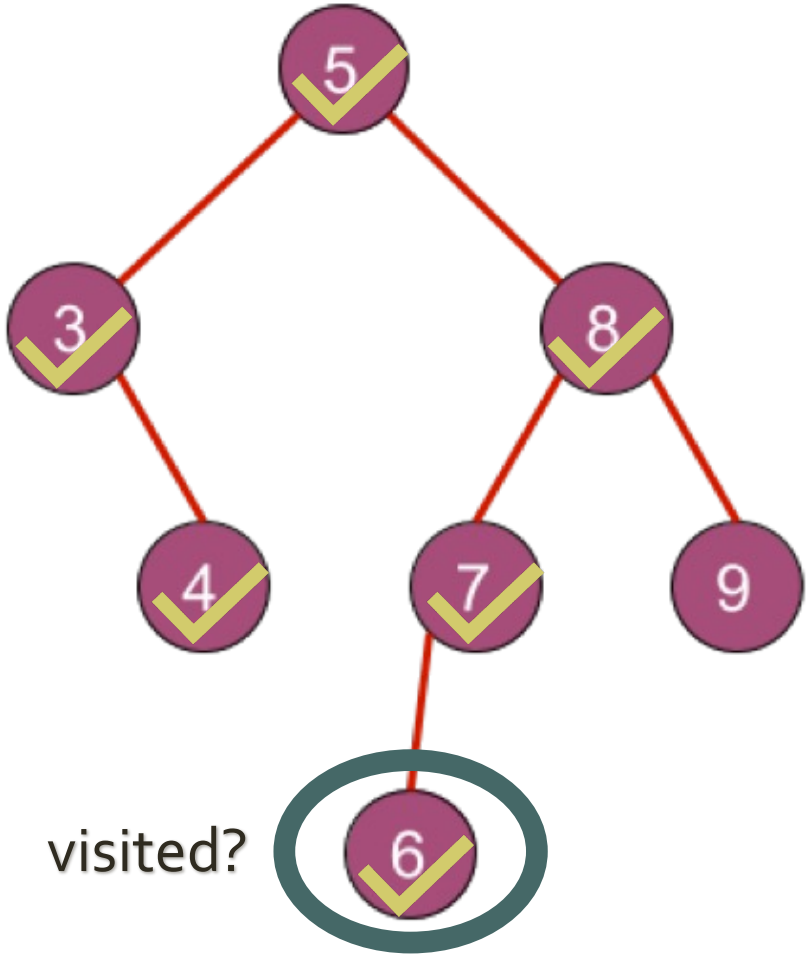
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



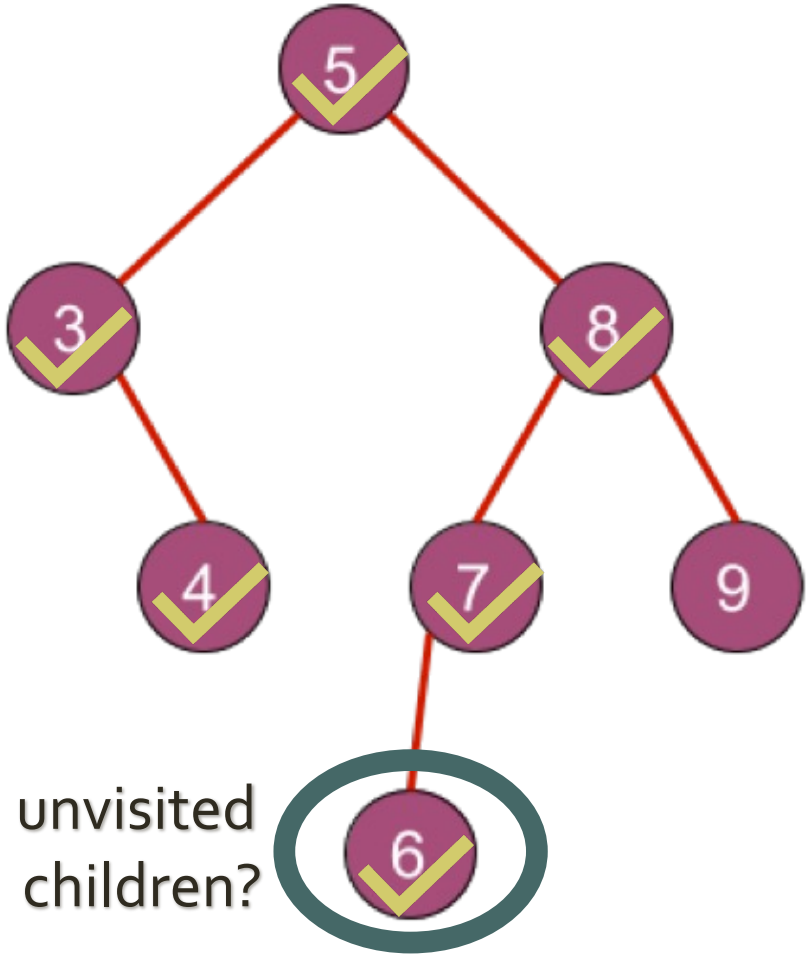
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



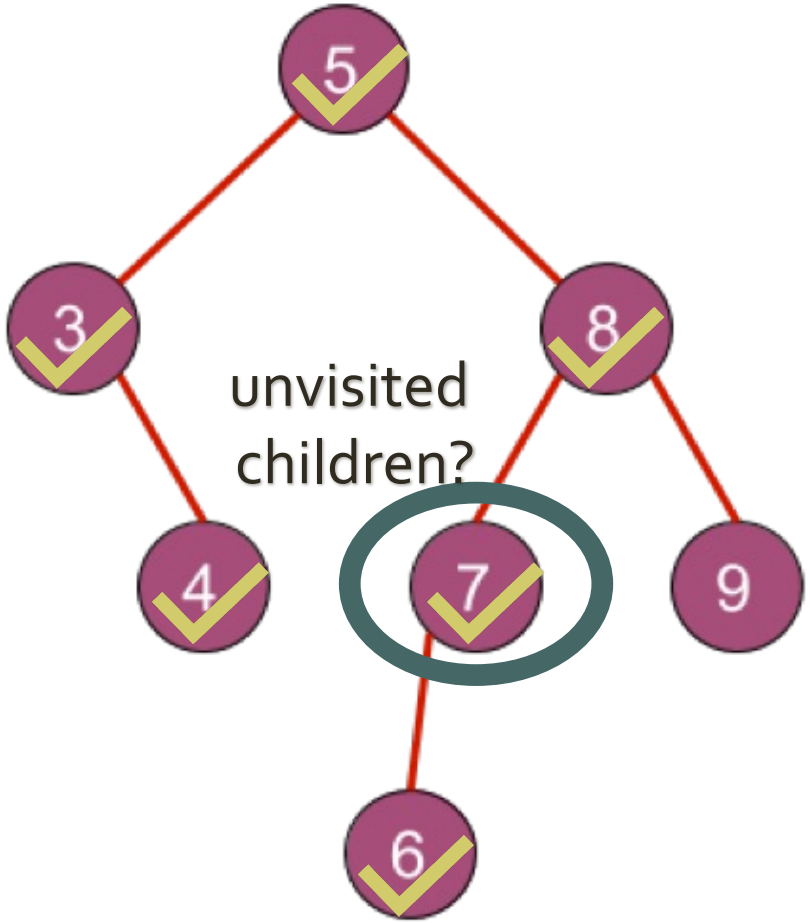
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



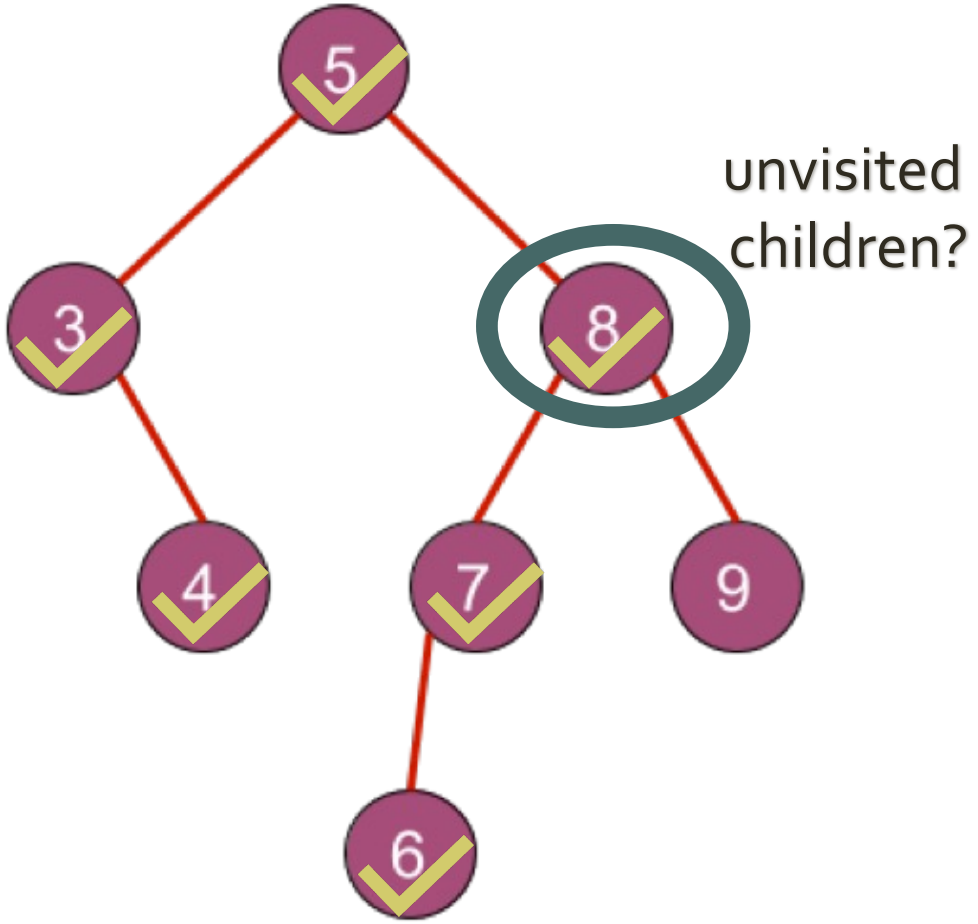
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



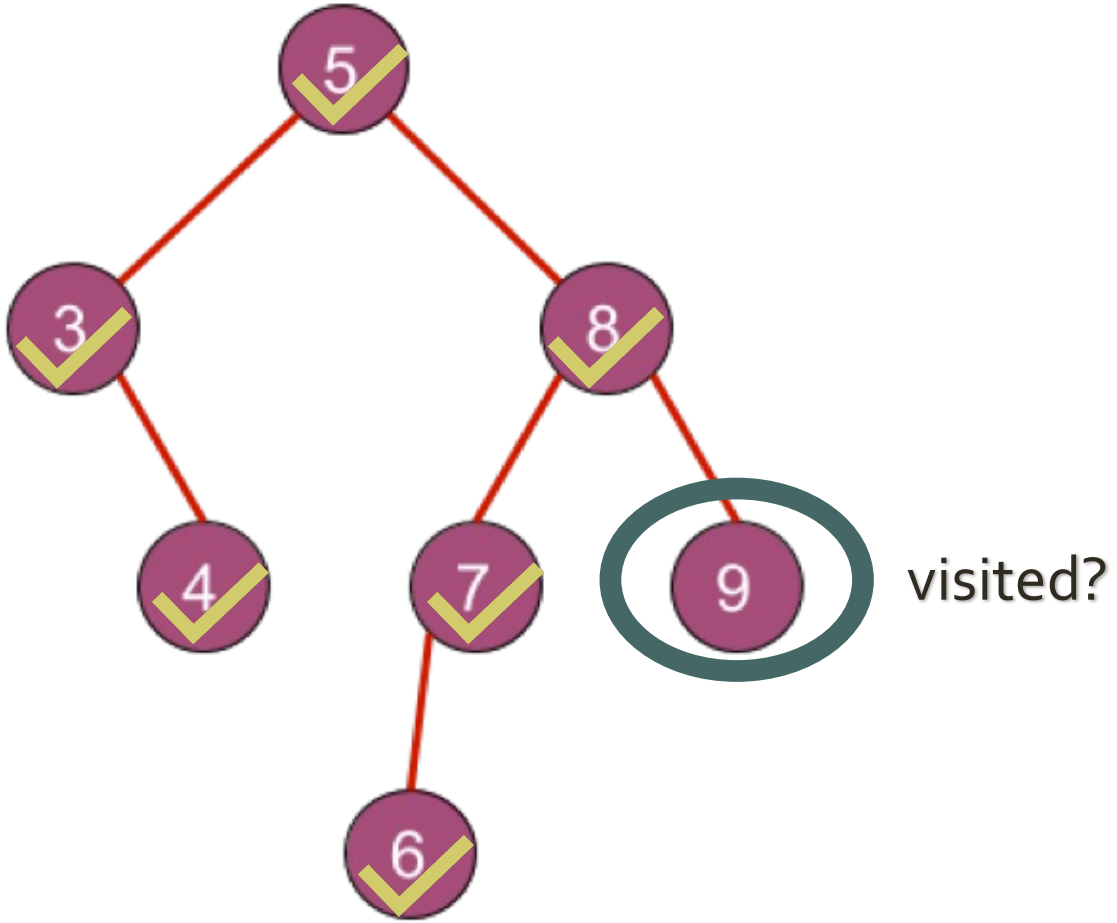
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



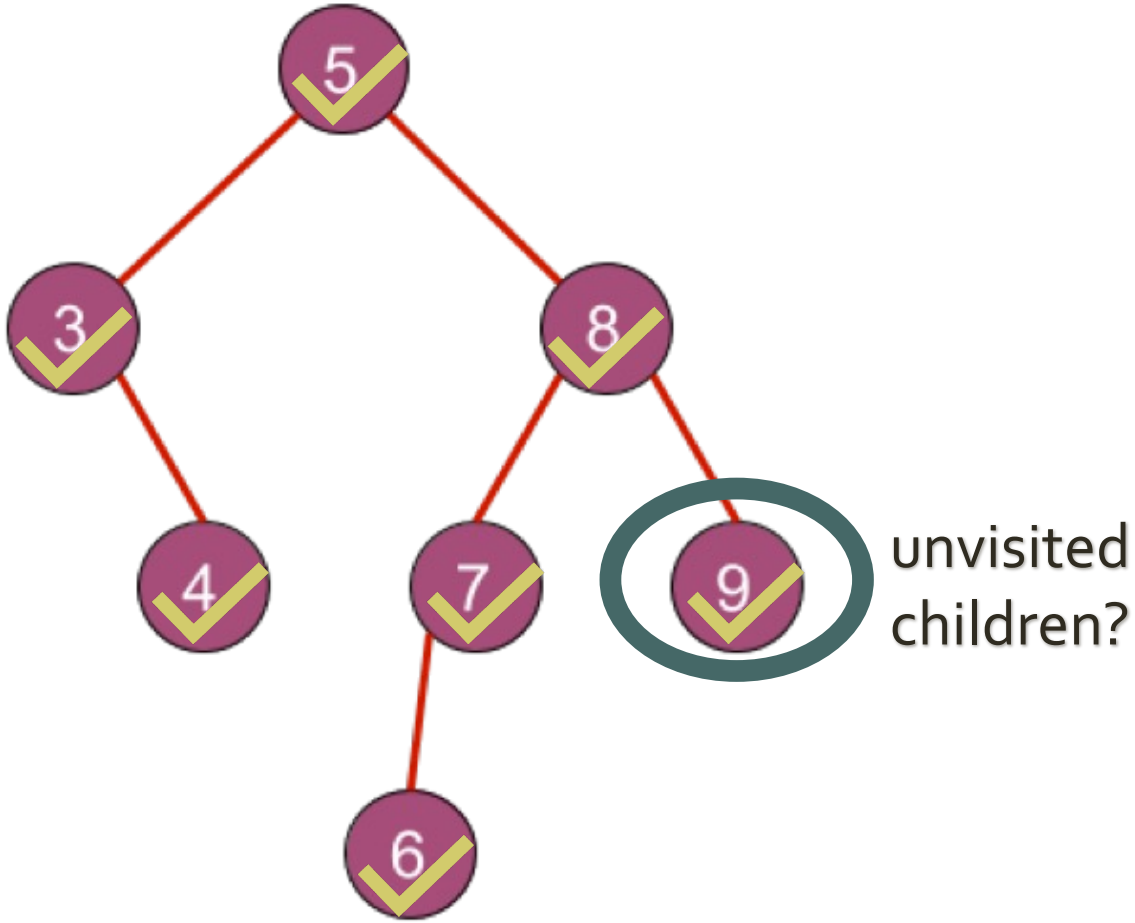
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



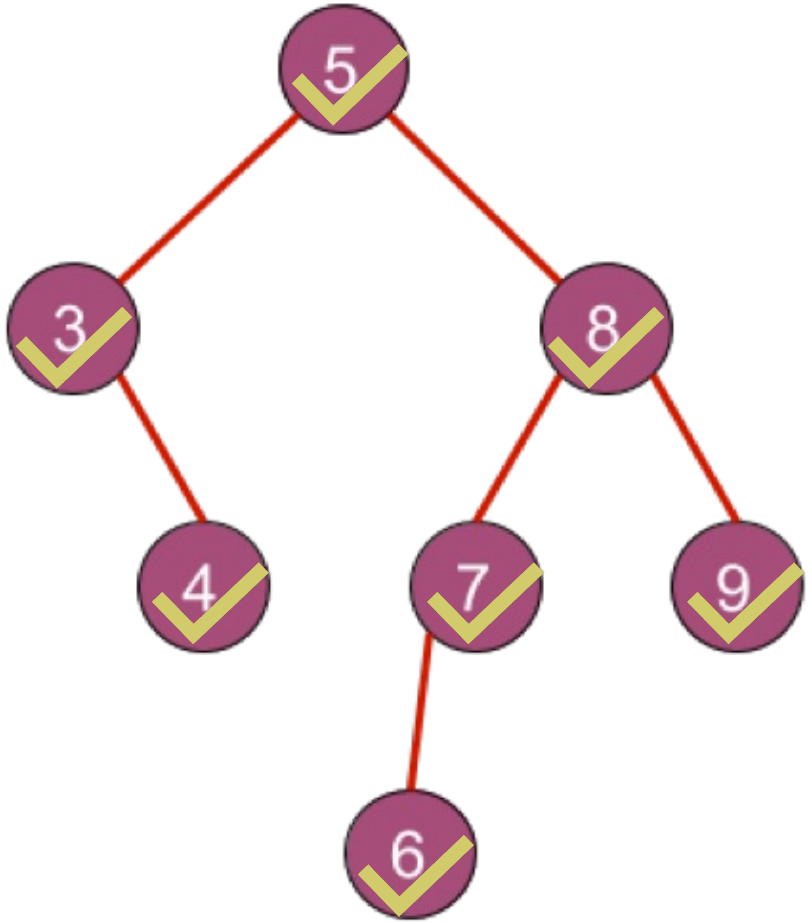
DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.



Done!

DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.

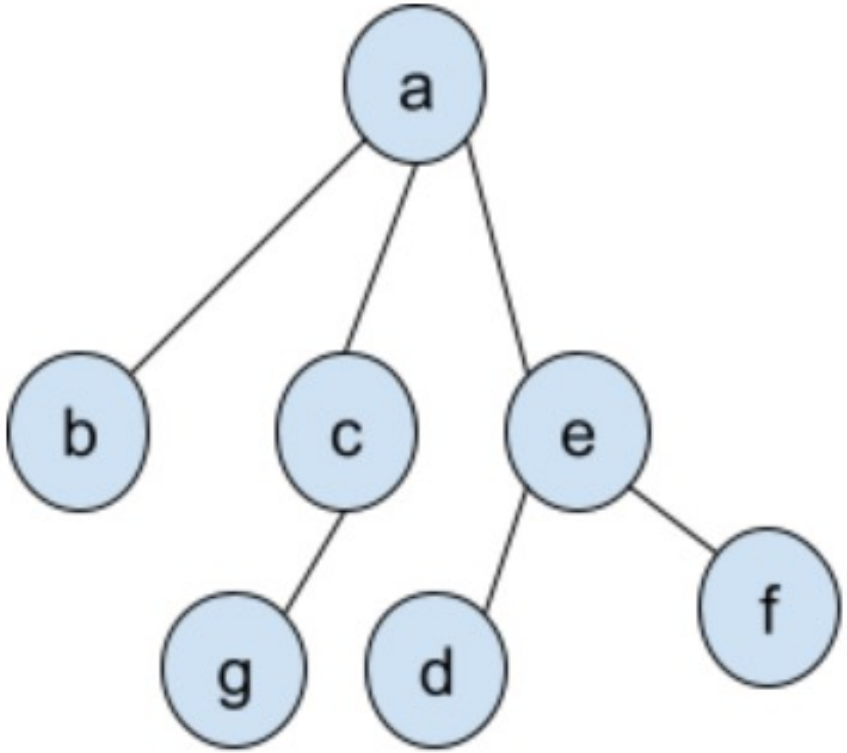
Algorithm:

```
DFS(G, u)
    mark u as visited
    for each v in G
        if v is not visited
                DFS(G, v)
```

DFS

In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.

Draw this graph on the board and do a DFS. Count how many steps the BFS takes.



DFS

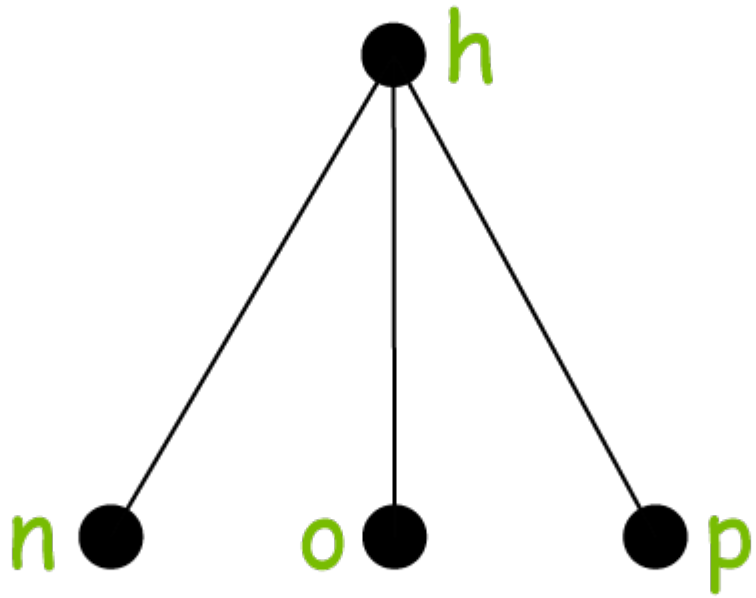
In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.

Draw this graph on the board and do a DFS. Count how many steps the BFS takes.

DFS

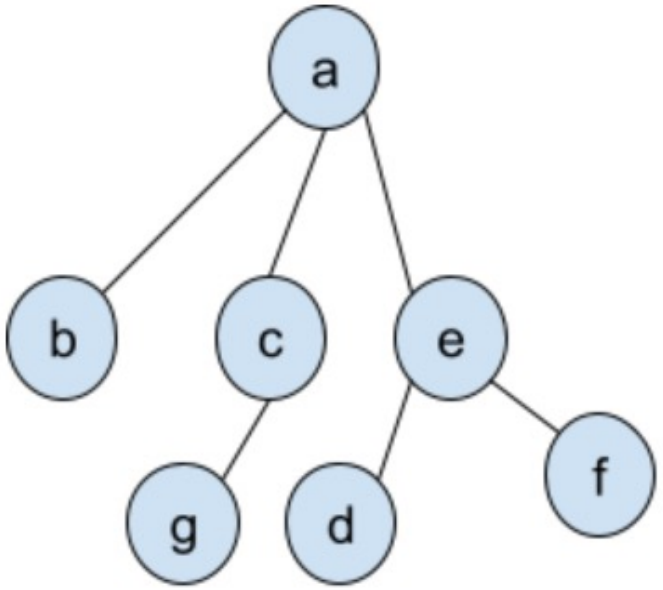
In a ***depth first search (DFS)*** we travel as far from the root as possible, then backtrack until we can move forward again.

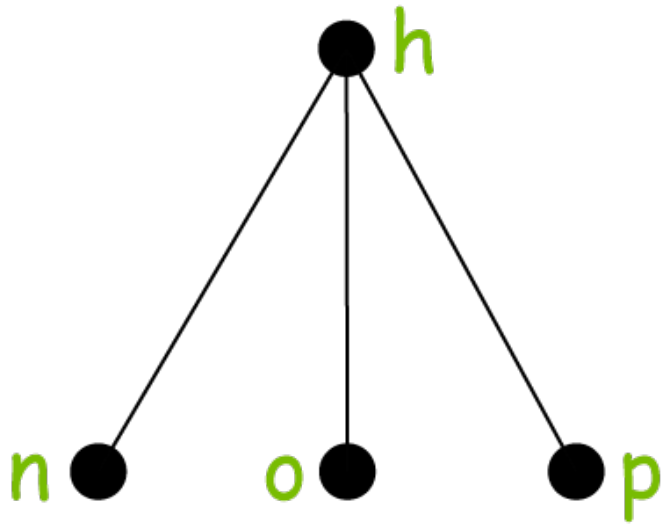
Draw this graph on the board and do a BFS. Count how many steps the DFS takes.



BFS vs. DFS

For each graph below, which took more steps, BFS or DFS?





BFS vs. DFS

For each graph below, which took more steps, BFS or DFS?

For each algorithm, what seems to impact number of steps?