# Graph Traversals

# Graph traversals

- Just like with trees, there is more than one to traverse (visit all the nodes of) a graph!
- Two common types of graph traversals:
  - **Breadth-first search** (BFS)
  - **Depth-first search** (DFS)

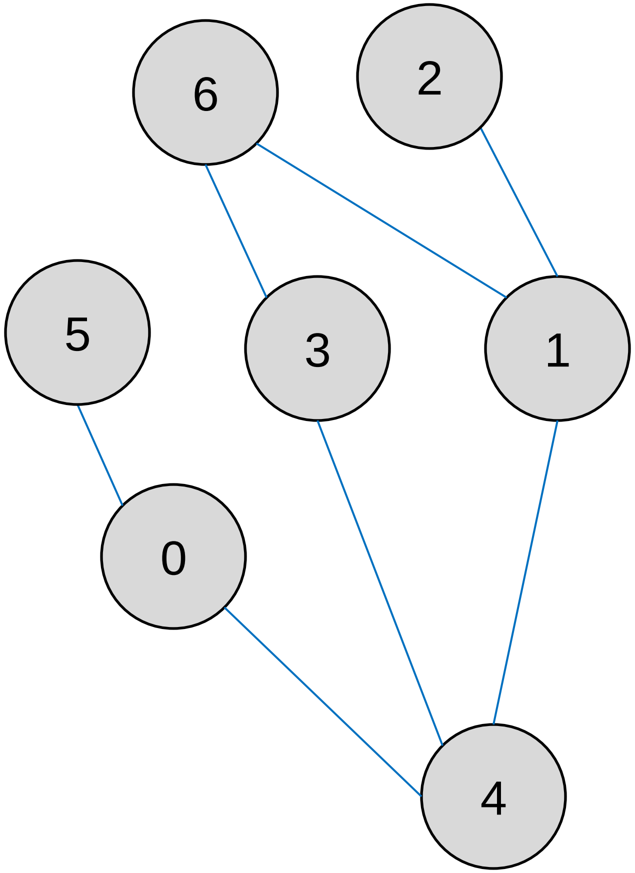
# Breadth-first search

- To perform a BFS starting from vertex *S*:
  - Mark vertex *S* as “identified”
  - Place vertex *S* into a queue
  - While there are vertices remaining in the queue:
    - Dequeue the next vertex, *V*
    - Visit vertex *V*
    - For each of *V*’s neighbors (adjacent vertices) that are both unvisited and unidentified:
      - Mark that neighbor as “identified”
      - Enqueue that neighbor into the queue
    - Mark vertex *V* as “visited”

# Breadth-first search: example

- Perform a BFS of this graph starting from vertex 4
  - Start by marking 4 as “identified” and placing it into the queue

| Dequeue and visit | Identified vertices | Queue | Visited vertices |
|---|---|---|---|
| 4 | 4 0 1 3 | 0 1 3 | 4 |
| 0 | 4 0 1 3 5 | 1 3 5 | 4 0 |
| 1 | 4 0 1 3 5 2 6 | 3 5 2 6 | 4 0 1 |
| 3 | 4 0 1 3 5 2 6 | 5 2 6 | 4 0 1 3 |
| 5 | 4 0 1 3 5 2 6 | 2 6 | 4 0 1 3 5 |
| 2 | 4 0 1 3 5 2 6 | 6 | 4 0 1 3 5 2 |
| 6 | 4 0 1 3 5 2 6 | (empty) | 4 0 1 3 5 2 6 |

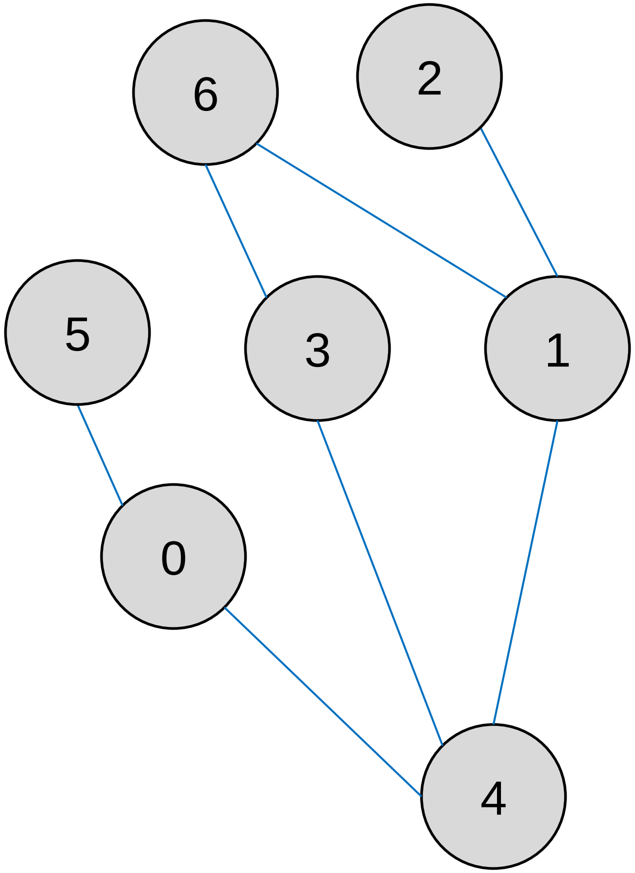


# Depth-first search

- To perform a DFS starting from vertex *S*:
  - Visit vertex *S*
  - Mark vertex *S* as “visited”
  - For each of *S*’s unvisited neighbors, perform a DFS starting from that neighbor

# Depth-first search: example

- Perform a DFS of this graph starting from vertex 4



| Call | Unvisited neighbors (recursive calls to DFS) | Visited vertices |
|---|---|---|
| DFS(4) | DFS(0), DFS(1), DFS(3) | 4 |
| DFS(0) | DFS(5) | 4 0 |
| DFS(5) | (none) | 4 0 5 |
| DFS(1) | DFS(2), DFS(6) | 4 0 5 1 |
| DFS(2) | (none) | 4 0 5 1 2 |
| DFS(6) | DFS(3) | 4 0 5 1 2 6 |
| DFS(3) | (none) | 4 0 5 1 2 6 3 |