



CIT 596

# Breadth-First Search

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# BREADTH-FIRST SEARCH

- Imagine that a large sound is made at the starting vertex.
- In what order would the sound wave reach the other vertices?
- Intuitively, this is order in which breadth-first search visits vertices.
- Algorithmically, BREADTH-FIRST-SEARCH( $v$ ) puts all neighbors of  $v$  in a queue
- It processes vertices in queue order.
- So neighbors of  $v$  will be visited before neighbors of neighbors of  $v$ .

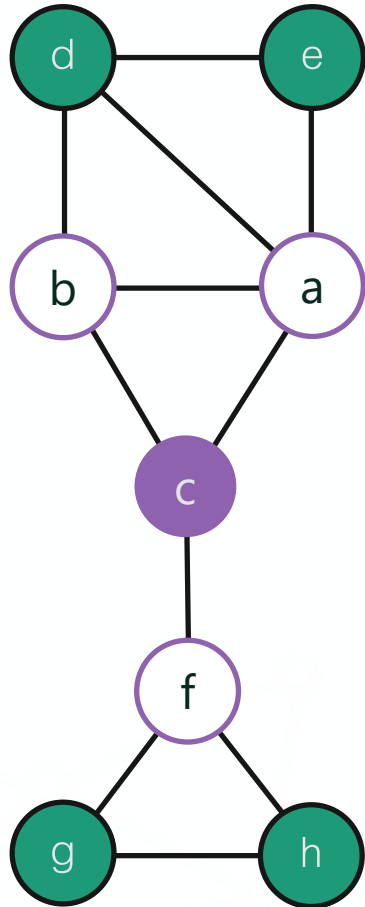
Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# BREADTH-FIRST SEARCH



- Vertices will be unseen, visited, or finished
- Key idea: If  $v$  is visited before  $u$ ,  $v$  will be finished before  $u$ . Use a queue to accomplish this.

Assignment Project Exam Help

- Visited = inserted into the queue

<https://tutorcs.com>

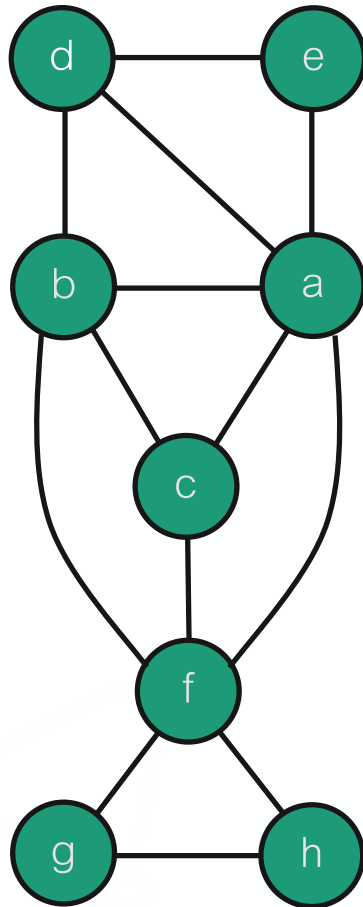
- Finished = all unseen neighbors added to the queue

- Figure: snapshot of BFS started on vertex c

WeChat: estutores

- Unseen = green
- Visited = white
- Finished = purple

# EXAMPLE



Assignment Project Exam Help

<https://tutors.com>

WeChat: cstutorcs

```
// all vertices start unseen
```

```
BREADTH-FIRST-SEARCH(G, s):
```

```
  initialize an empty queue Q
```

```
  mark s visited
```

```
  enqueue(Q, s)
```

```
  while Q is not empty
```

```
    x ← dequeue(Q)
```

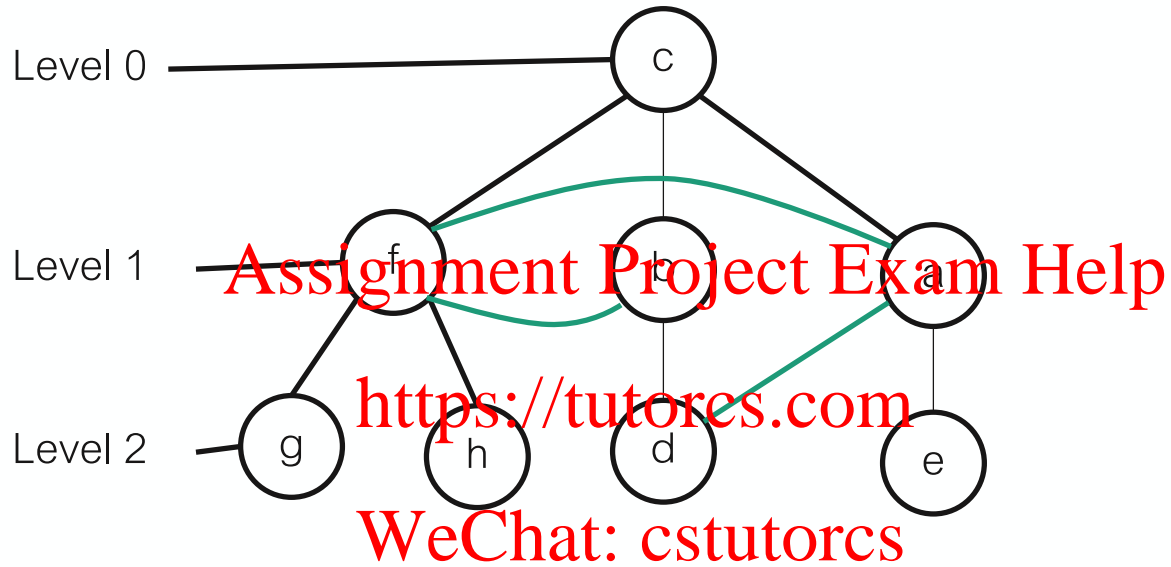
```
    for each unseen neighbor y of x
```

```
      mark y visited
```

```
      enqueue(Q, y)
```

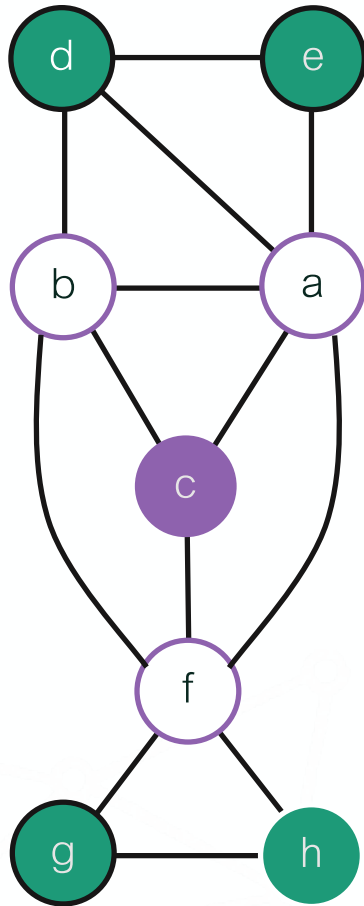
```
  mark x finished
```

# PROPERTIES OF BFS



- If  $u$  is enqueued while exploring  $v$  then  $\text{level}(u) = \text{level}(v) + 1$ .
- All non-tree edges are between vertices whose levels differ by at most 1.
- Vertices in a level are explored consecutively.
- Vertices are explored in order of levels.

# MORE PROPERTIES OF BFS



- $\text{level}(u)$  is the distance from  $c$  (the starting vertex) to  $u$ .
- Distance = number of edges in shortest path from  $c$  to  $u$ .
- Key application of BFS: shortest paths from source vertex

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# SHORTEST PATHS FROM NODE $v$

- Invoke BFS from node  $v$ .
- The level number of  $u$  tells us the length of the shortest path from  $v$  to  $u$ .
- Important point: here the length of a path is just the number of edges on it.
- Later, we will consider graphs with weights on the edges.
  - The total weight of a path is the sum of the weights of the edges on the path.
- BFS does not work for finding shortest paths in such graphs.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

