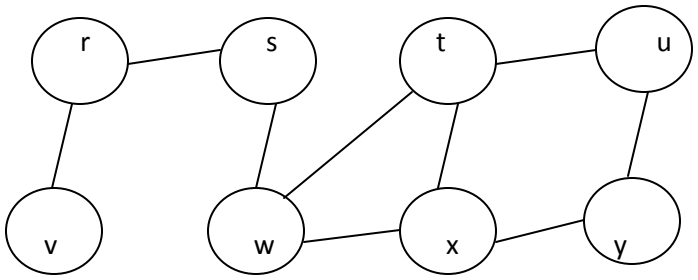
Graph Search Algorithms - I (BFS)

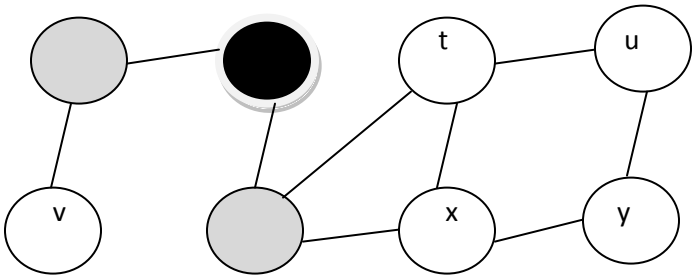
Graph search algorithms are the most common and widely used algorithms on graphs. They are used in real life situations like solving a maze, in compilers, routing etc.
Before moving to the algorithms I am assuming that you are clear about graphs and how they are implemented.
BFS:
The first algorithm I will talk about is called the breadth first search (BFS). It basically visits all the vertices in a connected component. The visit is done in levels (this will be clear as I progress). The visiting is done from a particular node and we call it a root. The data structure we need to implement BFS is a queue.



I will run BFS on this graph. Let the source be s and we put it in the queue. Whenever a vertex is enqueued it is coloured grey and when dequeued it will be coloured black and the unvisited ones are white. The colours are used for indications. As s is visited/enqueued first its level is 0.
Q = s
level (s) = 0
colour (s) = grey
pi/parent (s) = null
then a loop will run,
as long as the queue is not empty,
dequeue a vertex
colour it black
enqueue its unvisited neighbors (those coloured white)
colour them grey
their parent is the dequeued vertex
level them dequeued vertex level + 1

Q contains s and as it is not empty, dequeue s, colour it black and enqueue it unvisited neighbors w and r, colour them grey and parent of w and r is s. w and r will have levels 0 + 1 = 1.

Q = w,r
colour (s) = black
colour(w,r) = grey
pi/parent(w,r) = s
level(w,r) = 1



The loop continues until there are no more vertex to visit. At that point all the nodes will be black and all of them will have a level number. What does the level indicate ? It indicates when a vertex was visited and it also tells us the shortest distance from the source node. Parent tells us for which node it was enqueued plus we can also draw a tree from a graph based on the parent. Notice carefully that one vertex can be reached from multiple vertices but eventually one vertex has one parent. The parent vertex p of any vertex v, is that vertex that discovered v first when it was white. For the above graph w could have been reached from s, t and x but s is the parent because w was discovered by s first. Assigning one parent to any node eradicates cycles and hence a graph without a cycle is called a tree. A tree generated from a graph using BFS is called a BFS tree.

The overall algorithm of BFS is as follows:

```
for every vertex v in graph G
        colour[v] = white
        pi/parent[v] = null
        level[v] = -1
// let source vertex be denoted as s
```

```
enqueue s in queue Q
colour[s] = grey
level[s] = 0
while (Q not empty)
        d = dequeue[Q]
        for each neighbor vertex u of d
                if (color[u]==white)
                        enqueue[Q]=u
                        colour[u] = grey
                        level[u] = level[d]+1
                        pi/parent[u]=d
        colour[d]=black
```

Running time of BFS is O(V+E).