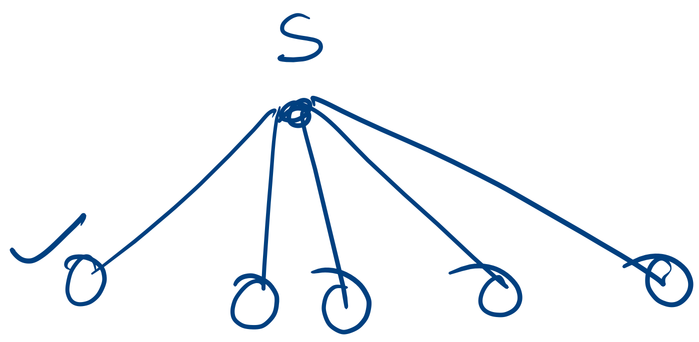
Graph Traversal Algorithms

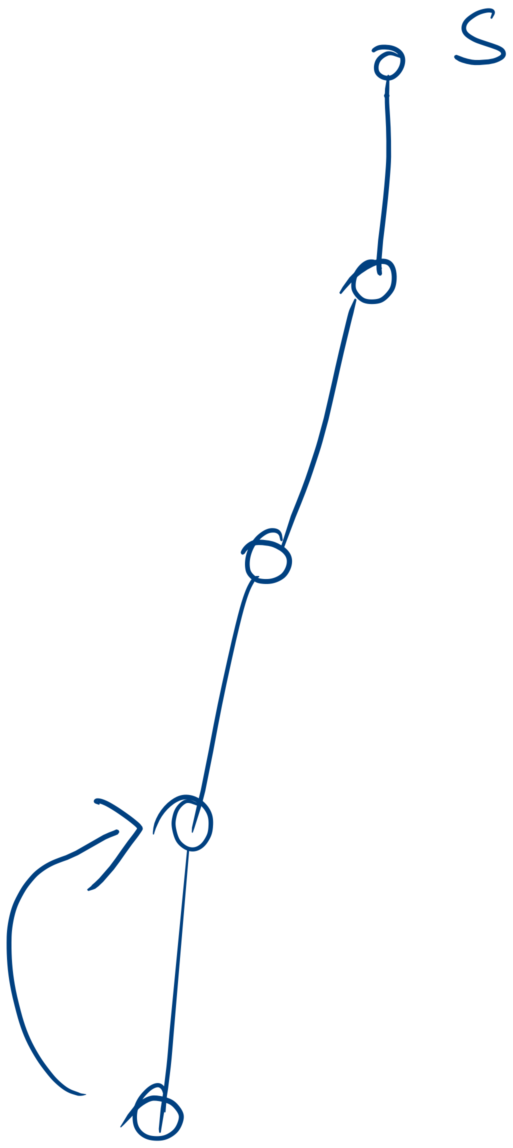
Breadth- First Search (BFS)

- Given a graph G(V,E) and a source vertex s, BFS "discovers" every vertex reachable from s

BFS



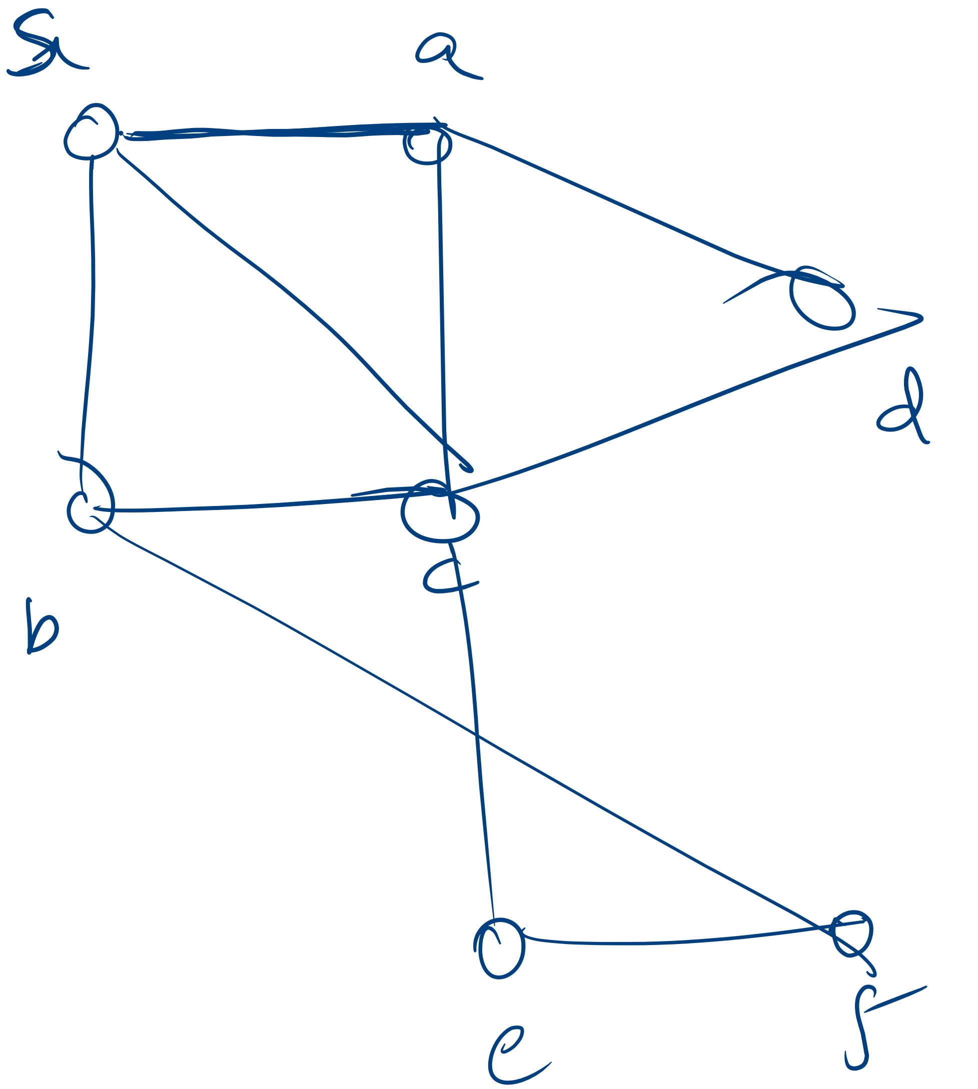
DFS



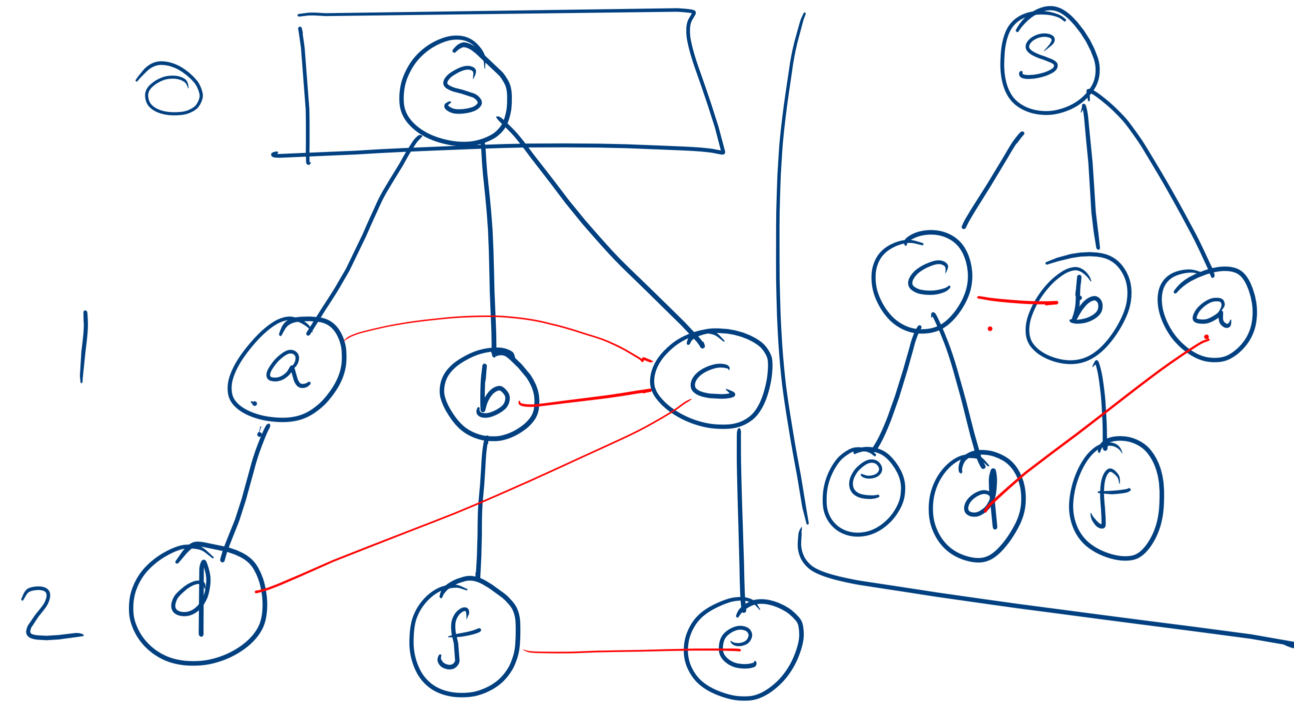
Breadth- First Search (BFS)

BFS colors vertices using 3 colours

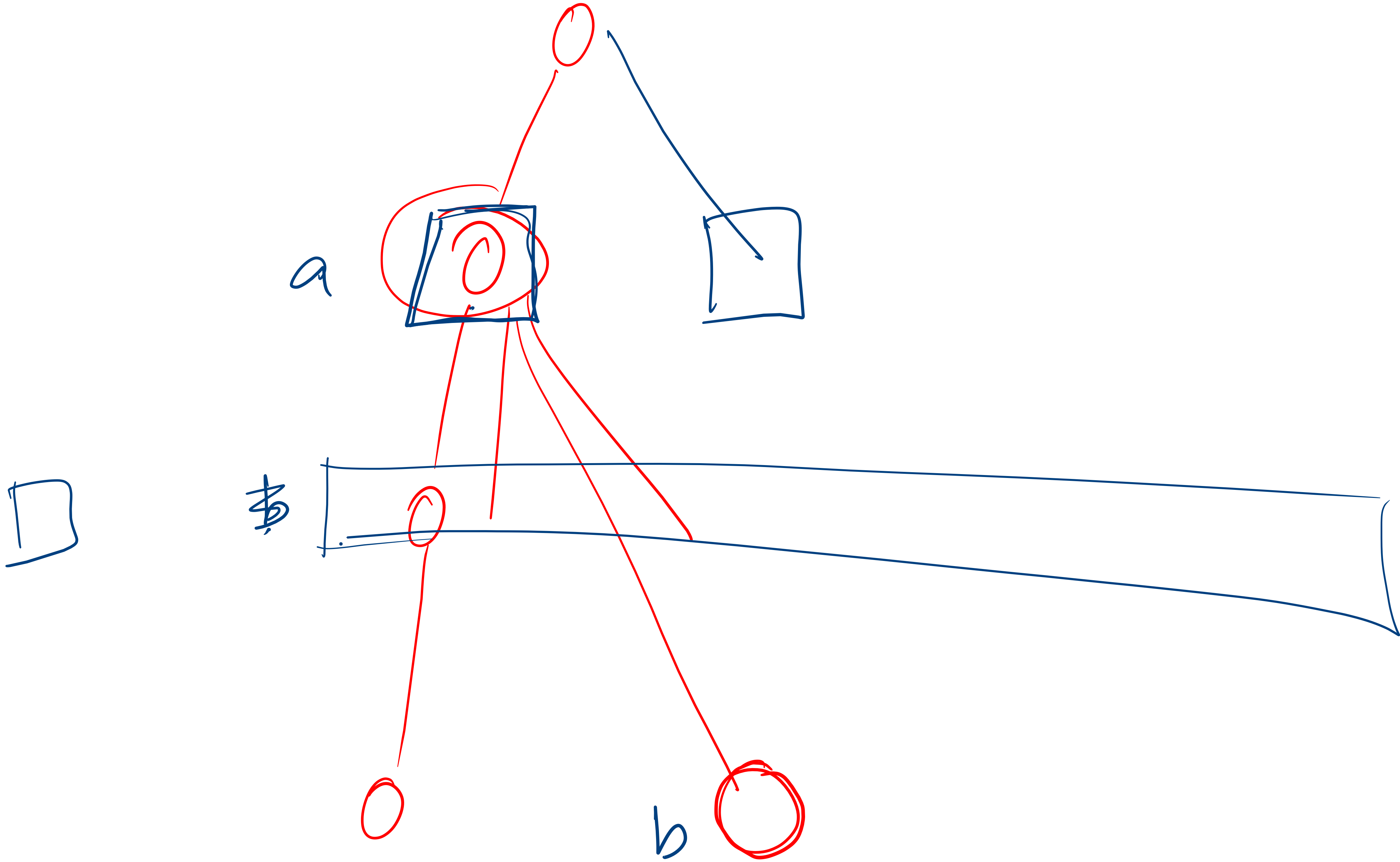
- white : a vertex not discovered
- gray : a discovered vertex with white neighbors
- black : a discovered vertex where every neighbour in either black or gray



s a b c d f e



Tree edges.


a
b

Breadth- First Search (BFS)

BFS Tree

Breadth- First Search (BFS)

BFS(G,s)

```
for each vertex u in v(G) -{s}
     color(u)= white
     d[u] = ∞
```

$\pi[u] = NIL$

```
color[s] = gray
d[s]=0
```

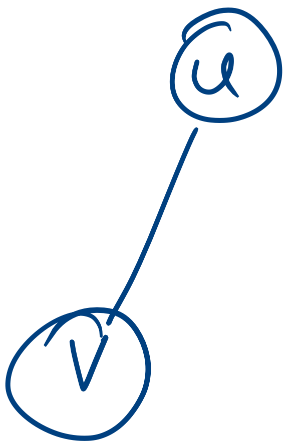
$\pi[s] = NIL$

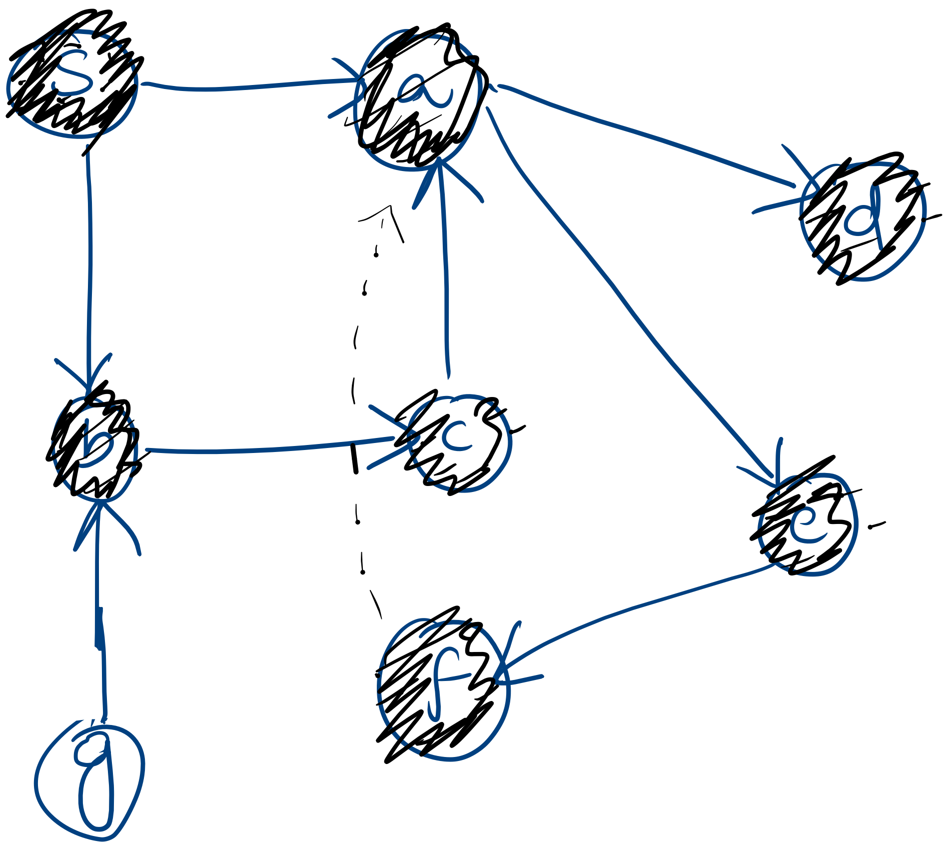
```
Q = ∅
```

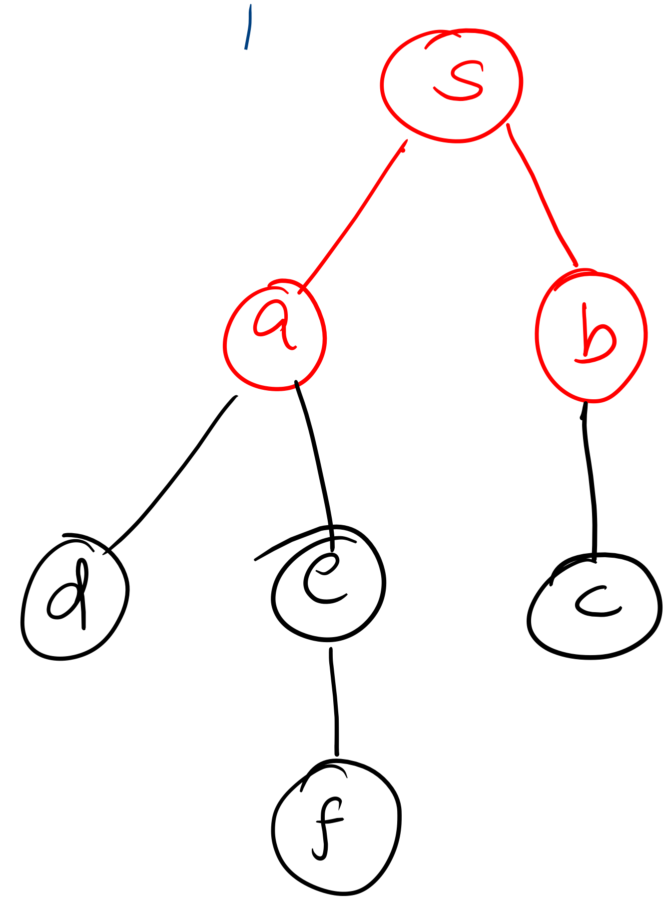
Breadth- First Search (BFS)

BFS(G,s)

```
Enqueue(Q,s)
while Q ≠ ∅
    u = Dequeue(Q)
    for v in Adj[u]
        if color[v] = white
            color[v] = gray
            d[v] = d[u]+1
            π[v] = u
            Enqueue(Q,v)
    color[u]=black
```







Q

~~a~~

~~b~~

~~d~~

~~e~~

~~c~~

f

Breadth- First Search (BFS)

Running Time ?

Breadth- First Search (BFS)

BFS(G,s)

```
for each vertex u in v(G) -{s}
     color(u)= white
     d[u] = ∞
```
$\pi[u] = NIL$

O(n)

```
color[s] = gray
d[s]=0
```
$\pi[s] = NIL$

```
Q = ∅
```

Breadth- First Search (BFS)

```
BFS(G,s)

Enqueue(Q,s)

while Q ≠ ∅

    u = Dequeue(Q)

    for v in Adj[u]

        if color[v] = white

            color[v] = gray

            d[v] = d[u]+1

            π[v] = u

            Enqueue(Q,v)

    color[u]=black
```

Every vertex is enqueued and dequeued once

Time taken for queue operations - O(n)

Breadth- First Search (BFS)

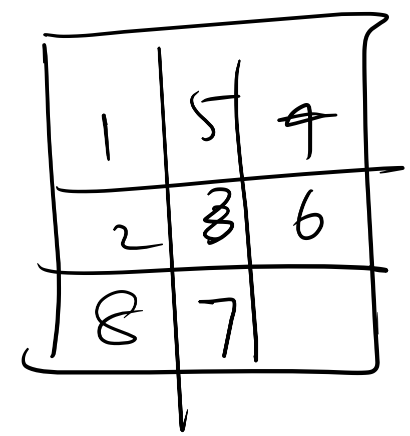
```
BFS(G,s)

Enqueue(Q,s)

while Q ≠ ∅

    u = Dequeue(Q)

    for v in Adj[u]

        if color[v] = white

            color[v] = gray

            d[v] = d[u]+1

            π[v] = u

            Enqueue(Q,v)

    color[u]=black
```

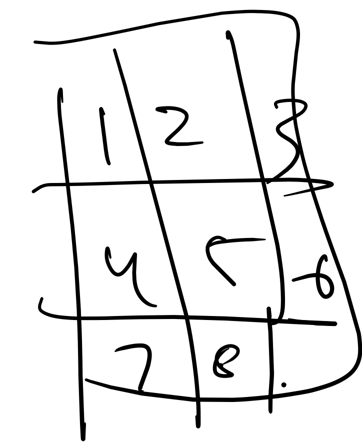
For a vertex dequeued, we visit every adjacent vertex

Time taken - O(sum of degrees) = O(m)

Breadth- First Search (BFS)

Running Time - O(n+m)





Breadth- First Search (BFS)

Properties of BFS :

Breadth- First Search (BFS)

Properties of BFS :

Shortest path :

`d[v] holds the value of the shortest path from s to v`

Breadth- First Search (BFS)

$\delta(u, v)$: shortest path from u to v

Let u,v in V[G] and (u,v) in E[G]

Then $\delta(s, v) \leq \delta(s, u) + 1$

Breadth- First Search (BFS)

$d[v] \geq \delta(s, v)$

Breadth- First Search (BFS)

$d[v] \geq \delta(s, v)$

Proof (by induction on #Enqueue operations)

Breadth- First Search (BFS)

Suppose that during the execution of BFS on a graph G(V,E), the queue Q contains $\{v_1, v_2, \ldots, v_r\}$ where v_1 is the head of Q and v_r is the tail. Then

$$d[v_i] \leq d[v_{i+1}]$$

$$d[v_r] \leq d[v_1] + 1$$

Breadth- First Search (BFS)

Suppose that during the execution of BFS on a graph G(V,E), the queue Q contains $\{v_1, v_2, \ldots, v_r\}$ where v_1 is the head of Q and v_r is the tail. Then

$$d[v_i] \leq d[v_{i+1}] \quad 1 \leq i \leq r-1$$

$$d[v_r] \leq d[v_1] + 1$$

Proof (by induction on queue operations):

$d(v_1) \leq (v_2) \leq (v_3) \ldots \leq d(v_r)$

$\underline{\# op^n s}$.

1 . Q contains only s. trivially true.

Claim holds for k queue $op^n s$.

Case 1) $(k+1)^{th}$ op^n is Dequeue.

$Q = \{v_1 \ldots \ldots v_r\}$.

dequeue v_1 is removed.

~~d~~ We know $d[v_r] \le d[v_1] + 1 \le d[v_2] + 1$.

To show $d[v_r] \le d[v_2] + 1$.

2) Enqueue opⁿ.

u, v_1, ..., v_r, v_{r+1}.

let u is the parent of v_{r+1}.

u is already ~~enq~~ dequeued.

$$d[u] \le d[v_1]$$

$$d[v_{r+1}] = d[u] + 1$$

$$\le d[v_1] + 1$$

$$d[v_r] \le d[v_{r+1}]$$

$$d[u] \le d[v_1]$$

$$\Rightarrow d[v_r] \le d[u] + 1$$

$$= d[v_{r+1}]$$

Breadth- First Search (BFS)

Suppose that during the execution of BFS on a graph G(V,E), the queue Q contains $\{v_1, v_2, \ldots, v_r\}$ where v_1 is the head of Q and v_r is the tail. Then

$$d[v_i] \leq d[v_{i+1}]$$

$$d[v_r] \leq d[v_1] + 1$$

Corollary : Let v_i, v_j be enqueued during the execution of BFS such that v_i is enqueued before v_j. Then $d[v_i] \leq d[v_j]$

Breadth- First Search (BFS)

Suppose that during the execution of BFS on a graph G(V,E), the queue Q contains $\{v_1, v_2, \ldots, v_r\}$ where v_1 is the head of Q and v_r is the tail. Then

$$d[v_i] \leq d[v_{i+1}]$$

$$d[v_r] \leq d[v_1] + 1$$

Corollary : Let v_i, v_j be enqueued during the execution of BFS such that v_i is enqueued before v_j. Then $d[v_i] \leq d[v_j]$

Breadth- First Search (BFS)

BFS discovers every vertex that is reachable from s and upon termination $d[v] = \delta(s, v)$.

Moreover, for $v \neq s$ that is reachable from s, one of the shortest path from s to v, is a shortest path from s to $\pi[v]$, followed by ($\pi[v]$,v).

$d[v]$

$\delta(s,v)$ = length of shortest path from s to v.

Claim: $d[v] = \delta(s,v)$

$d[v] \geq \delta(s,v)$

---

let v be s.t. $\underline{d[v] > \delta(s,v)}$, and v is the nearest vertex to s

uv is the last edge in the shortest path b/w s and v. u is dequeued.

$d[v] > \delta(s,v) = \delta(s,u) + 1$

$= d[u] + 1$

$d[v] > \underline{d[u]} + 1.$

v cannot be black.

v - white $\Rightarrow d[v] = d[u]+1.$

v - grey

wv $\in$ BFS tree

$d[v] = d[w]+1$

$\leq d[u]+1.$