

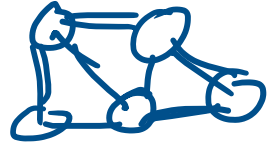
$V = \text{total vertices}$

$E = \text{total edges}$ $|E|$

BFS-Visit(G, s) // search graph G starting from vertex s .

001

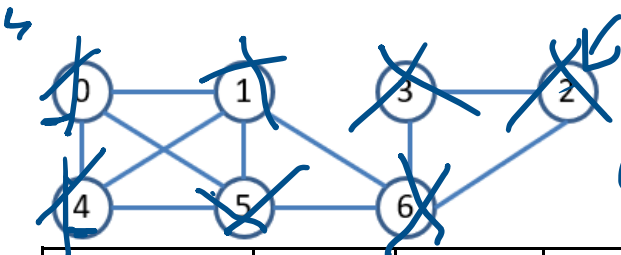
1. For each vertex u of G
 1. $\color{red}{\cancel{color[u] = WHITE}}$ // undiscovered // mandatory
 2. $\color{red}{\cancel{dist[u] = inf}}$ // distance from s to u // optional
 3. $\color{red}{\cancel{pred[u] = NIL}}$ // predecessor of u on the path from s to u // optional $NIL \rightarrow -1$
2. $\color{red}{\cancel{color[s] = GRAY}}$ // s is being processed
3. $\color{red}{\cancel{dist[s] = 0}}$
4. $\color{red}{\cancel{pred[s] = NIL}}$
5. Initialize empty queue Q
6. $\color{red}{\cancel{put(Q, s)}}$ // s goes to the end of Q
- 7. While Q is not empty $\rightarrow O(V)$
 1. $u = \text{get}(Q)$ // removes u from the front of Q $u: 2 \ 0(1)$
 2. For each v adjacent to u // explore edge (u, v) // in increasing order $\rightarrow O(V)$
 1. If $color[v] == WHITE$
 1. $color[v] = GRAY$
 2. $dist[v] = dist[u] + 1$
 3. $pred[v] = u$
 4. $put(Q, v)$ $O(1)$
 3. $color[u] = BLACK$



for each vertex run loop of $O(\text{degree}(u)) \Rightarrow O(E)$

Representation	BFS time complexity	BFS space complexity
<u>Adj LIST</u>	$O(E + V)$	$O(V)$
<u>Adj MATRIX</u>	$O(V^2)$	$O(V)$

correction



BFS($6, 2$)

$s=2$
 \emptyset white
 \emptyset gray
 \emptyset black

$Q: 2, 3, 6, 1, 5, 0, 4$
 $u: 2, 3, 6, 1, 5, 0, 4$

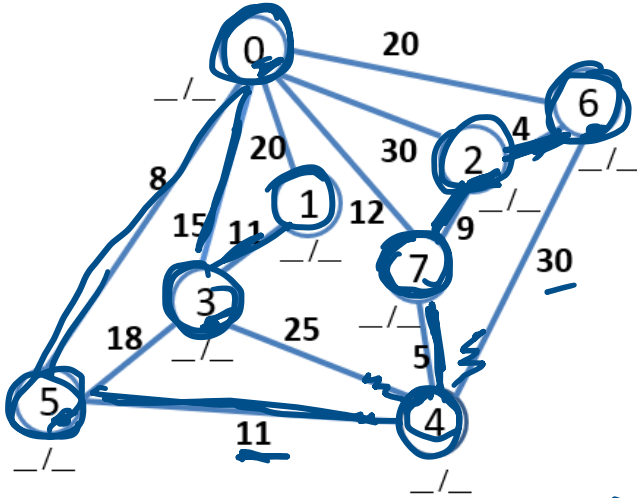
Vertex	0	1	2	3	4	5	6
Work (dist and parent / pred updates for nodes) dist / pred	$3/1$	$2/6$	$0/-1$	$1/2$	$3/1$	$2/6$	$1/2$


```

MST_Prim(G,w,s) // N = |V|
1  int d[N], p[N]
2  For v = 0 -> N-1
3    d[v]=inf //min weight of edge connecting v to MST
4    p[v]=-1 //MST vertex, s.t. w(p[v],v) =d[v]
5  d[s]=0
6  Q = PriorityQueue(G.V, d) //
7  While notEmpty(Q)
8    u = removeMin(Q,d) //add to tree //
9    for each v adjacent to u //u -> v
10   if v in Q and w(u,v) <= d[v] {
11     p[v]=u
12     d[v] = w(u,v);
13     decreasedKeyFix(Q,v,d) //v is neither index nor key //
14   }
15 }

```

Assume adjacency list representation. TC: SC:



MST(G, 7)

u: (0,∞) / (1,∞) ... (7,0)

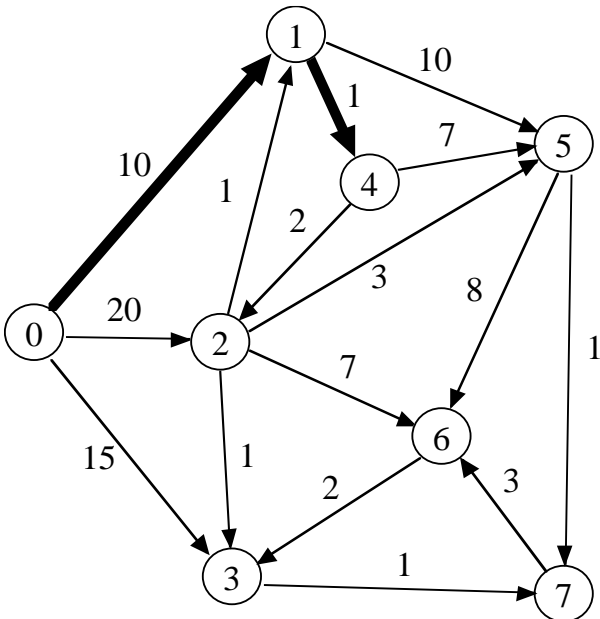
Vertex	0 X	1	2 X	3 X	4 X	5 X	6 X	7 X
Work (dist and parent updates for nodes) dist/parent	∞/-1 12/7 not put 30/2 S.C. 50/12 8/5 X	∞/-1 20/0 11/3 X	∞/-1 9/7 X	∞/1 25/4 18/5 15/0 X	∞/1 5/7 X	∞/-1 11/4 X	∞/-1 30/4 4/2 X	∞/-1 0/1 X

Dijkstra(G,w,s) // N = |V|

```

1  int d[N], p[N]
2  For v = 0 -> N-1
3    d[v]=inf //total weight from s to v
4    p[v]=-1 //v's predecessor on path s to v
5  d[s]=0
6  Q = PriorityQueue(d)
7  While notEmpty(Q) {
8    u = removeMin(Q,w)
9    for each v adjacent to u
10     if v in Q and (d[u]+w(u,v))<d[v]{
11       p[v]=u
12       d[v] = d[u]+w(u,v);
13       decreasedKeyFix(Q,v,d)
14     }
15 }

```



Vertex	0	1	2	3	4	5	6	7
Work (dist and parent updates for nodes)								