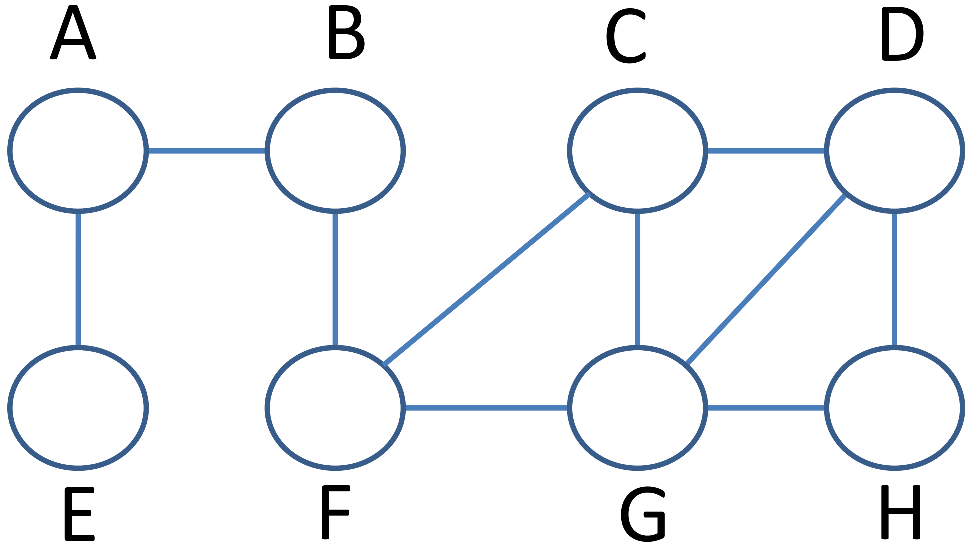
# BREADTH-FIRST SEARCH (G, s)

```
for each vertex u ∈ V [G]-s
        do color[u] ← WHITE
           distance[u] ← ∞
           predecessor[u] ← NIL
color[s] ← GRAY
distance[s] ← 0
predecessor[s] ← NIL
Q ← ∅
ENQUEUE (Q, s)
while Q ≠ ∅
        do u ← DEQUEUE (Q)
            for each v ∈ Adj[u]
                do if color [v] = WHITE
                        then color [v] ← GRAY
                             distance[v] ← distance[u] + 1
                             predecessor[v] ← u
                             ENQUEUE (Q, v)
            color[u] ← BLACK
```

# BREADTH-FIRST SEARCH

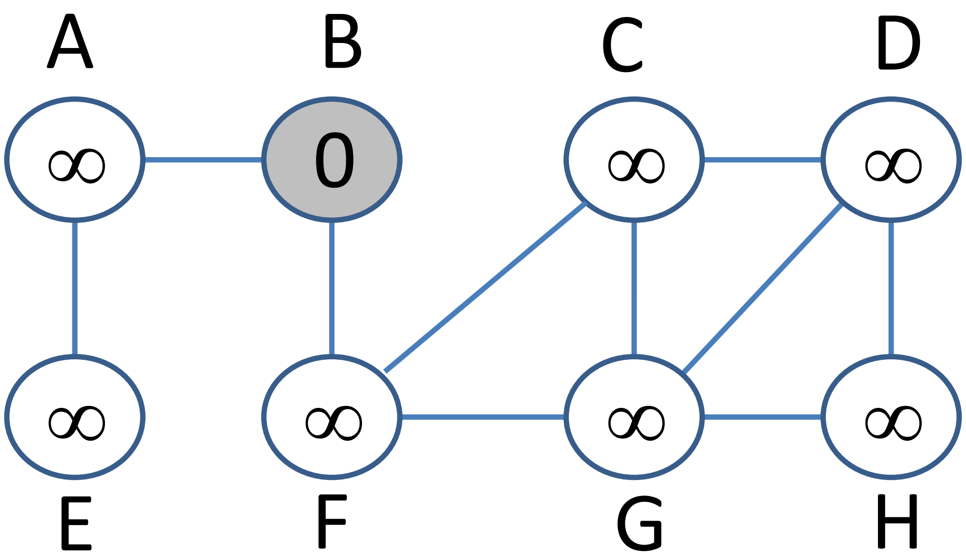
**Source** s = B

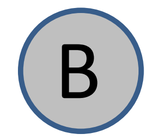


# BREADTH-FIRST SEARCH

**<u>Source</u>** s = B

**Breadth-first tree**



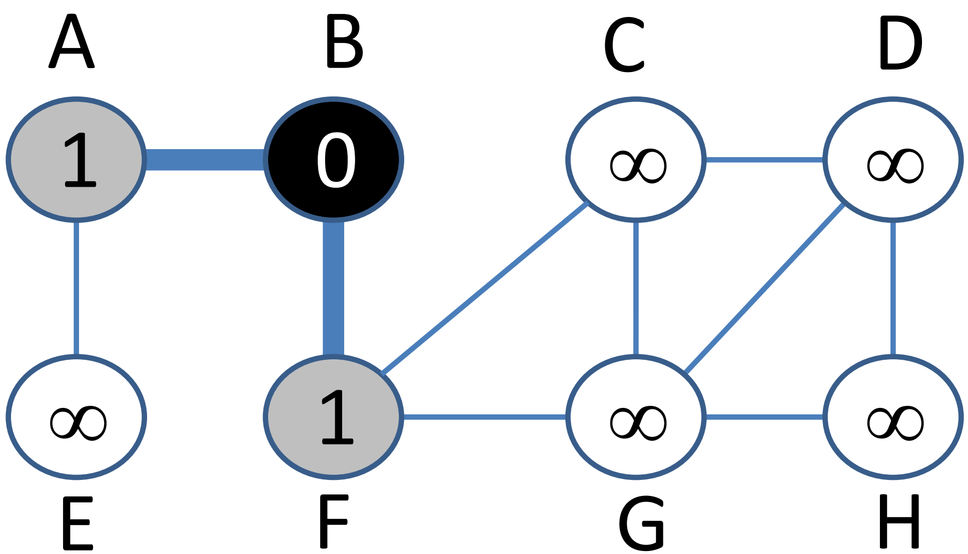

$\mathbf{Q = \{B_0\}}$

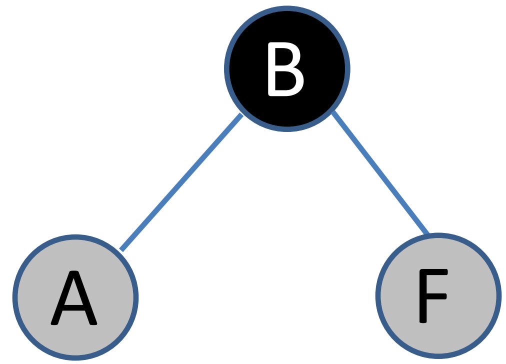
# BREADTH-FIRST SEARCH

$Q = \{B_0\}$

**expand B**

Breadth-first tree



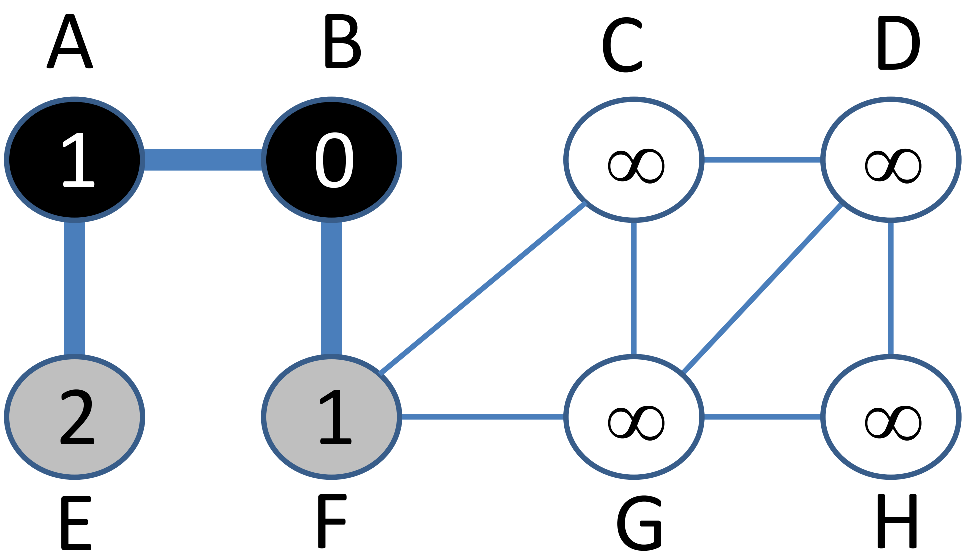

$\mathbf{Q = \{A_1, F_1\}}$

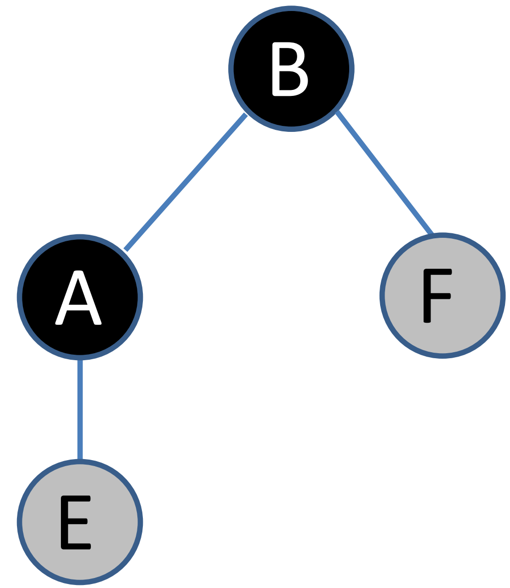
# BREADTH-FIRST SEARCH

$Q = \{A_1, F_1\}$

**expand A**

Breadth-first tree



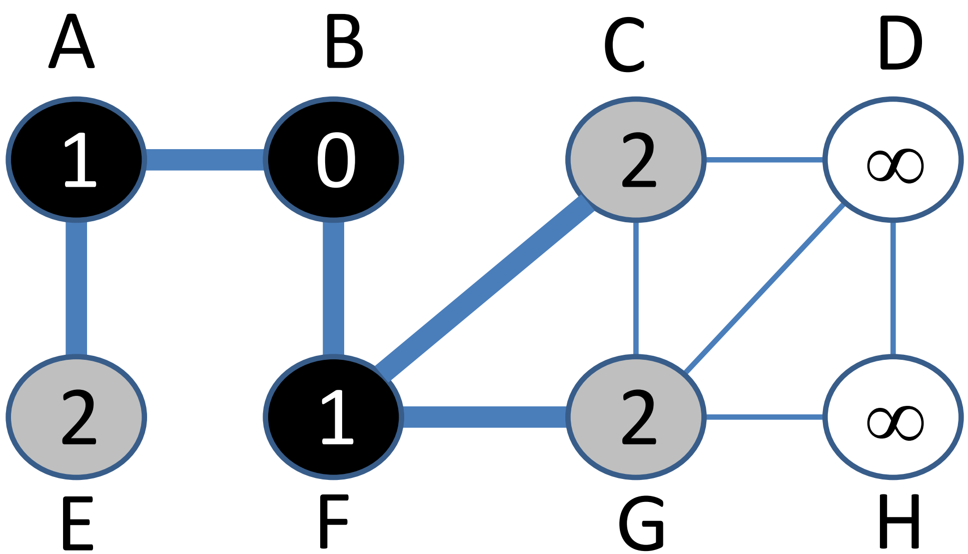

$\mathbf{Q = \{F_1, E_2\}}$

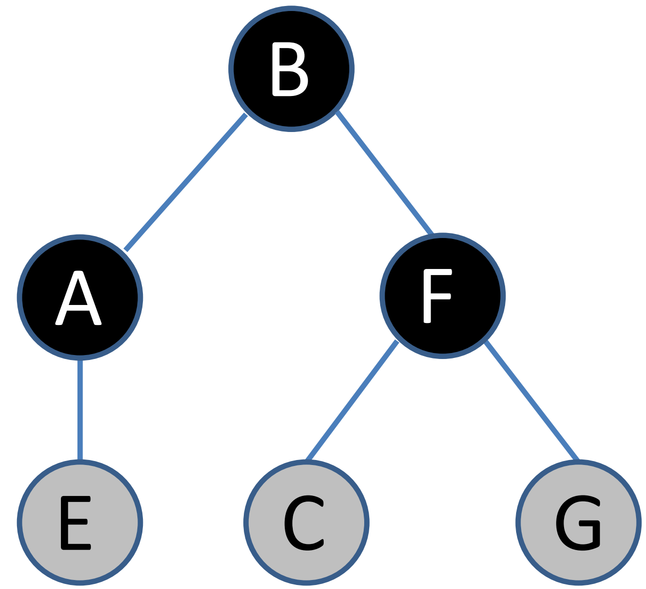
# BREADTH-FIRST SEARCH

$Q = \{F_1, E_2\}$

**expand F**

Breadth-first tree

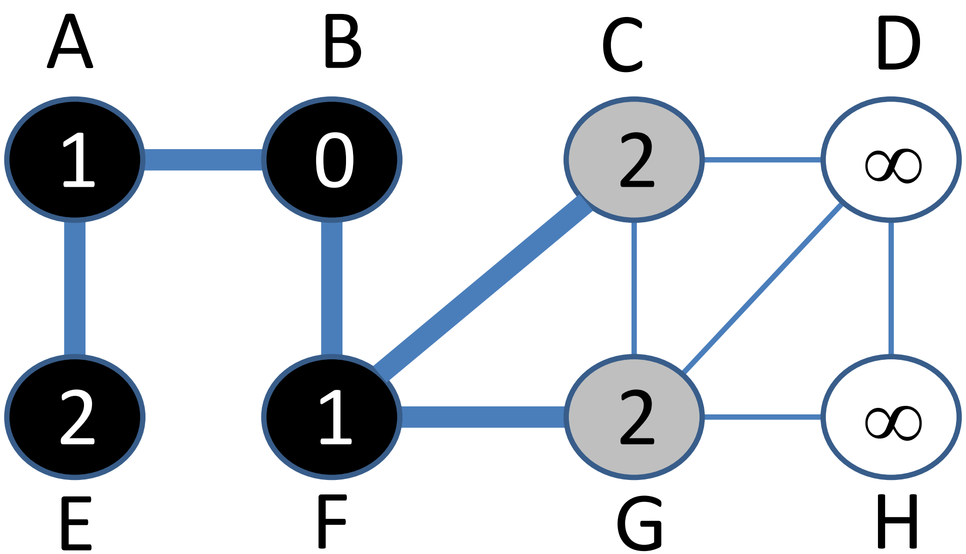


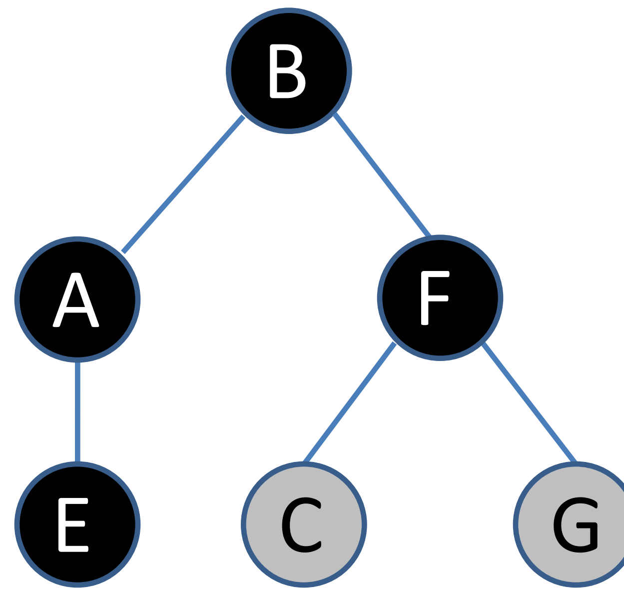




**$Q = \{E_2, C_2, G_2\}$**

# BREADTH-FIRST SEARCH

$Q = \{E_2, C_2, G_2\}$ **expand E** Breadth-first tree



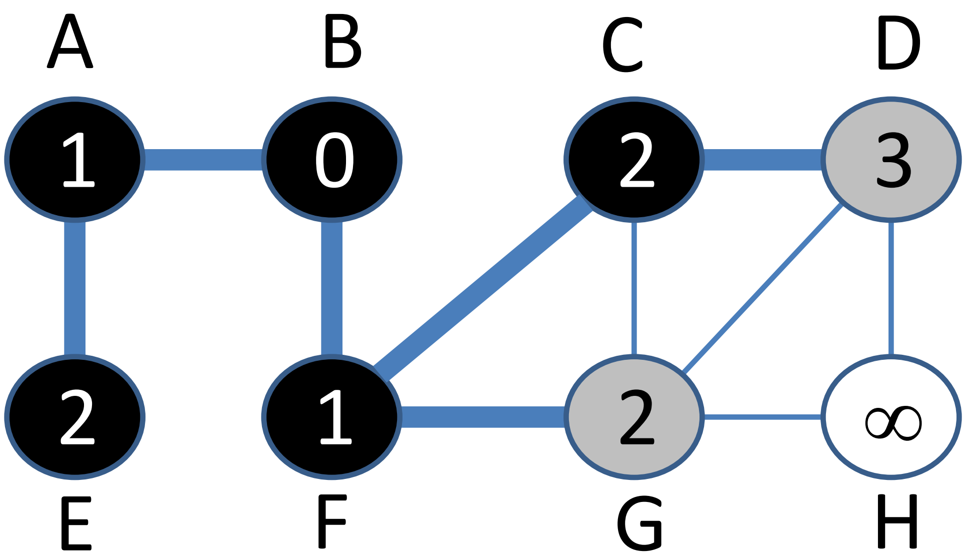

$\mathbf{Q = \{C_2, G_2\}}$

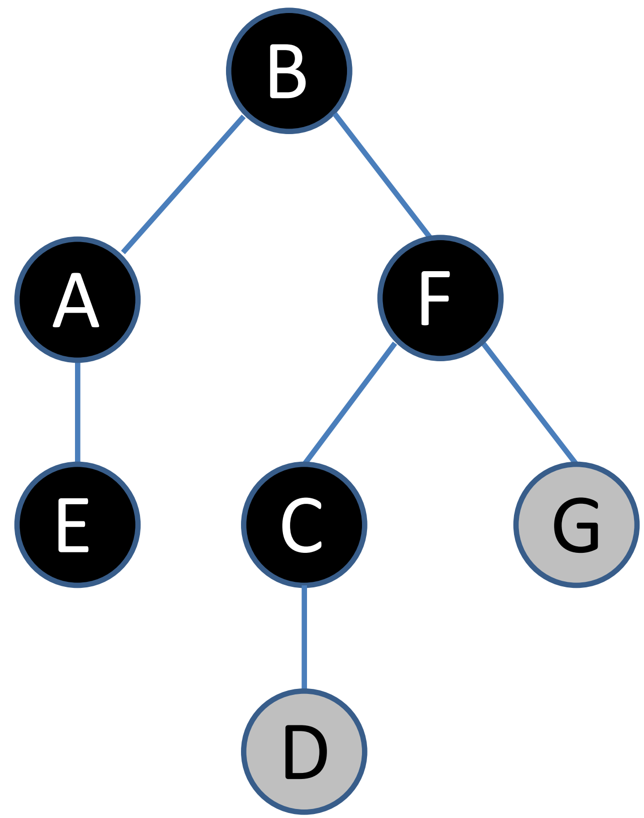
# BREADTH-FIRST SEARCH

$Q = \{C_2, G_2\}$

**expand C**

Breadth-first tree

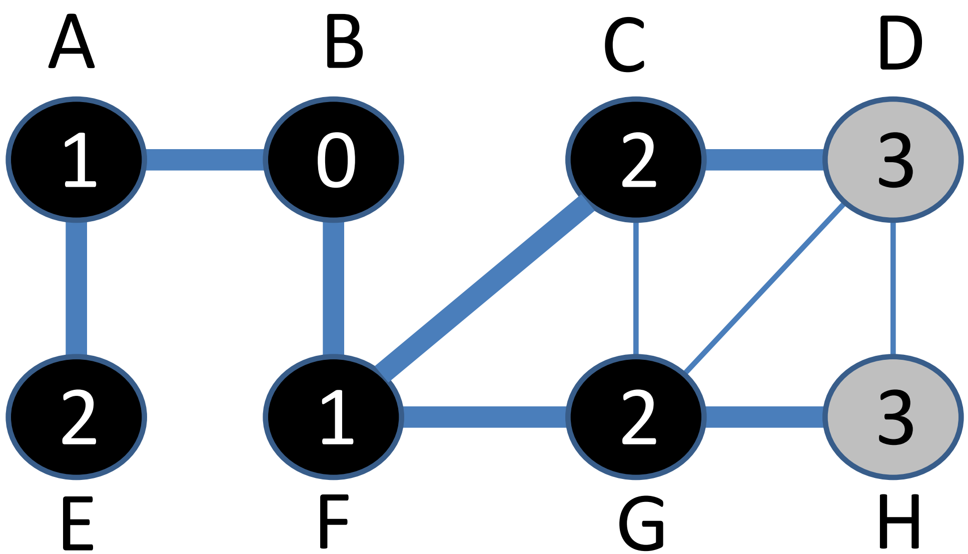


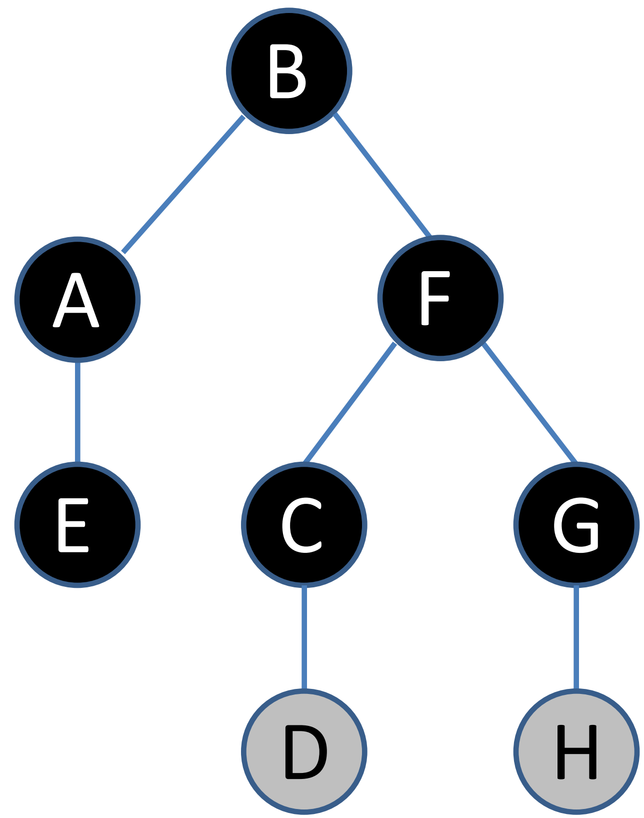




**$Q = \{G_2, D_3\}$**

# BREADTH-FIRST SEARCH

$Q = \{G_2, D_3\}$ **expand G** Breadth-first tree



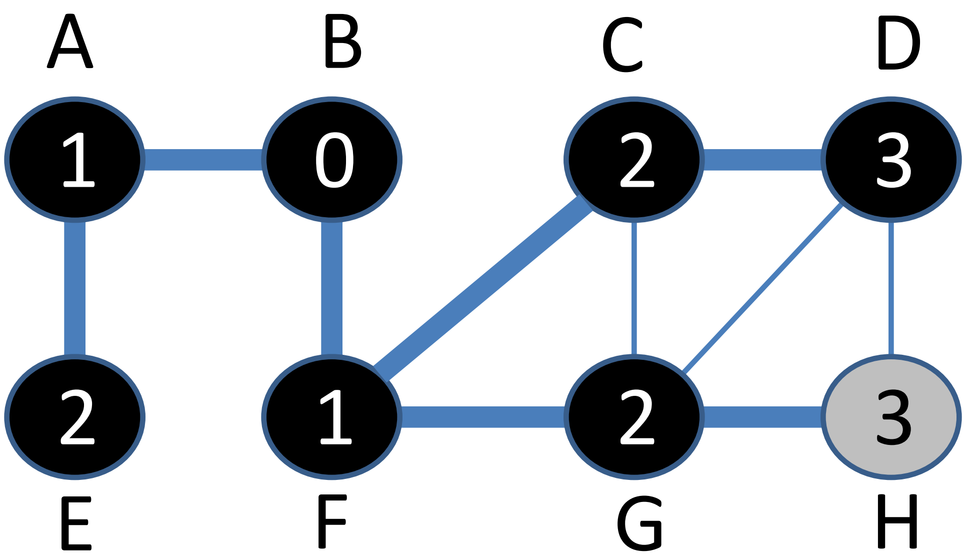

$\mathbf{Q = \{D_3, H_3\}}$

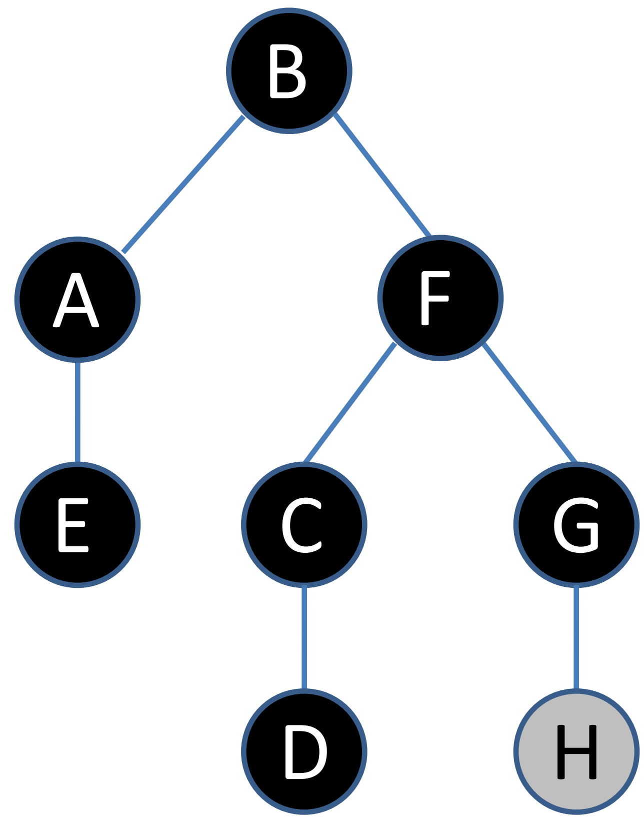
# BREADTH-FIRST SEARCH

$Q = \{D_3, H_3\}$

**expand D**

Breadth-first tree



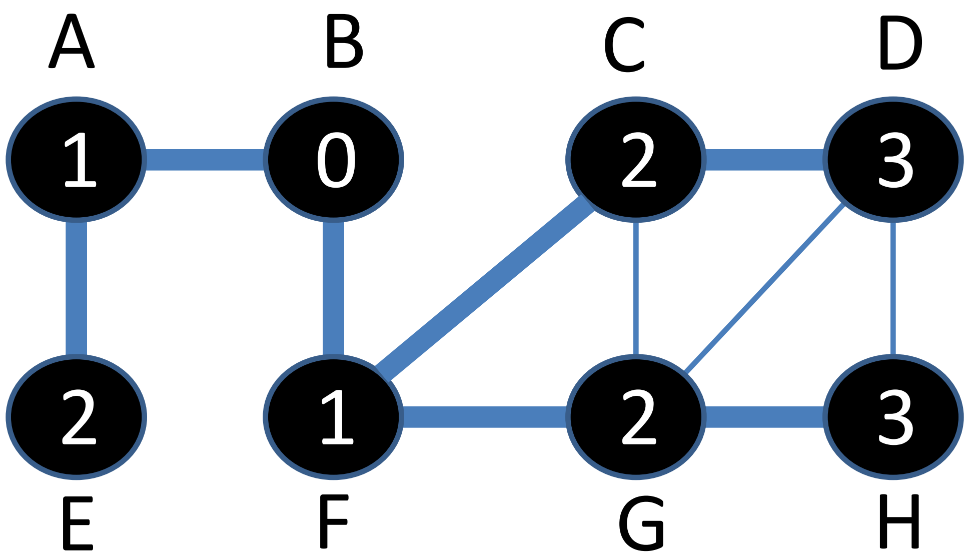




$\mathbf{Q = \{H_3\}}$

# BREADTH-FIRST SEARCH

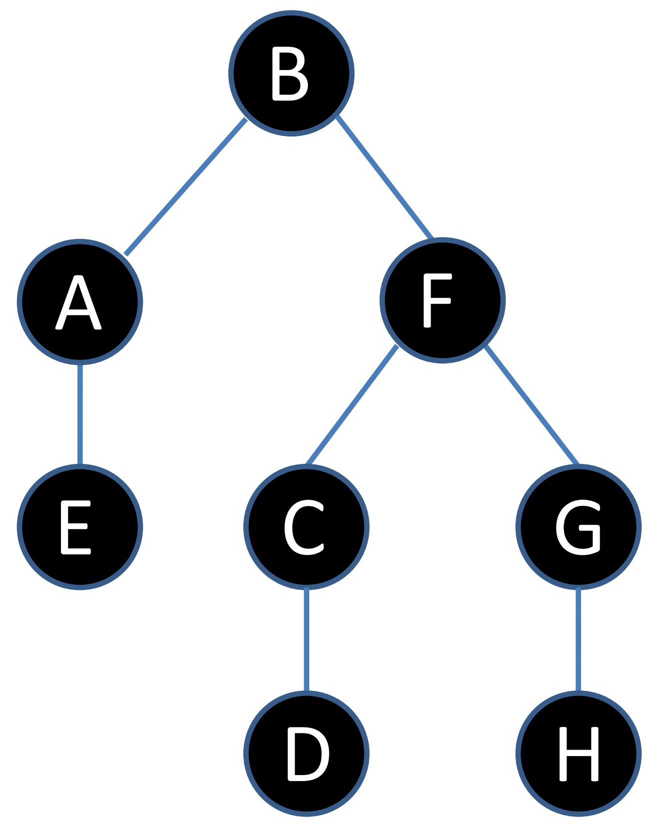
$Q = \{H_3\}$

**expand H**




**$Q = \{\}$**

Breadth-first tree



# Proofs of the properties of BFS

1. BFS finds the shortest path between two nodes

2. BFS  is correct

3. BFS finds the BF tree

# Proofs of the properties of BFS

1. **BFS finds the shortest path between two nodes**

2. BFS is correct

3. BFS finds the BF tree

# Shortest paths

$\delta(s,v)$=: minimum number of edges to reach v from s

(the graph is not weighted)

$\delta(s,v) = \infty$, if v cannot be reached from s

**L1:** For every edge (u,v): $\delta(s,v) \leq \delta(s,u) + 1$

**Dem:**

# Shortest paths

**<u>L1:</u>** For every edge (u,v): $\delta(s,v) \leq \delta(s,u) + 1$

**Dem:**

If $\exists$ trajectory (s,u), to reach v from s, one would need at most one more edge: the edge (u,v)

otherwise, $\delta(s,v) = \infty$, and the inequality also obtains

# Shortest paths

**L2:** After having completed the BFS algorithm,

$\forall$ v: **distance[v]≥ $\delta$(s,v)**

**Dem (induction):**

# Shortest paths

**<u>L2:</u>** After having completed the BFS algorithm,

$\forall$ v: **distance[v]≥ $\delta$(s,v)**

**Dem (induction):**

*Base case:* initially **d[s]= 0 = $\delta$(s,s) y d[v]= ∞ ≥ $\delta$(s,v)**

*hypothesis*: **d[u]≥ $\delta$(s,u)**

*induction:* v has been found from u (v is "white"), then:

$$\begin{aligned} d[v] &= d[u]+1 && \text{(line 15)} \\ &\geq \delta(s,u) + 1 && \text{(hypothesis)} \\ &\geq \delta(s,v) && \text{(L1)} \end{aligned}$$

Once v is ENQUEUED, d[v] does not change.

# Shortest paths

**<u>L3:</u>** Q contains ($v_1$, $v_2$…$v_r$). Then:

**$d[v_r] \leq d[v_1]+1$**, and **$d[v_i] \leq d[v_{i+1}]$,** for i=1, 2,… r-1

**Dem (induction on the number of operations in Q):**

# Shortest paths

**L3(1/2):** Q contains ($v_1$, $v_2$…$v_r$). Then:

**$d[v_r] \leq d[v_1]+1$**, and **$d[v_i] \leq d[v_{i+1}]$,** for i=1, 2,… r-1

**Dem (induction on the number of operations in Q):**

*Base case:* it is true initially (s is the only node in Q)

*Induction 1:* (DEQUEUE $v_1$) $\Rightarrow$ the fist vertex in Q is $v_2$

$d[v_1] \leq d[v_2]$ (hypothesis)

$$d[v_r] \leq d[v_1]+1$$
$$\leq d[v_2] +1$$

The remaining inequalities for the other vertices do not change

# Shortest paths

**L3:** Q contains ($v_1$, $v_2$…$v_r$). Then:

**$d[v_r] \leq d[v_1]+1$**, and **$d[v_i] \leq d[v_{i+1}]$,** for i=1, 2,… r-1

**Dem (induction on the number of operations in Q):**

*Base case:* it is true initially (s is the only node in Q)

*Inducción 2:* (ENQUEUE $v_{r+1}$)

# Shortest paths

<u>**L3 (2/2):**</u> Q contains ($v_1$, $v_2$…$v_r$). Then:

**$d[v_r] \leq d[v_1]+1$**, and **$d[v_i] \leq d[v_{i+1}]$,** for i=1, 2,… r-1

**Dem (induction on the number of operations in Q):**

*Base case:* it is true initially (s is the only node in Q)

*Induction 2:* (ENQUEUE $v=v_{r+1}$) $\Rightarrow$ “u” has been eliminated from Q. We are exploring vertex v (adjacent to u). There is a new $v_1$.

$d[u] \leq d[v_1]$ (hypothesis)

$d[v_{r+1}] = d[v] = d[u] + 1 \leq d[v_1]+1$ (v is adjacent to u)

$d[v_r] \leq d[u] +1$ (hypothesis)

Therefore: $d[v_r] \leq d[u] +1 = d[v] = d[v_{r+1}]$

The remaining inequalities for the other vertices do not change.

# Proofs of the properties of BFS

1. BFS finds the shortest path between two nodes

2. **<u>BFS is correct</u>**

3. BFS finds the BF tree

# Correctness of BFS

**Theorem:** BFS on G, from s. Then

- During the execution, BFS discovers all the vertices that are reachable from s
- At the end of the algorithm, d[v]= $\delta$(s,v) for all v
- For each v≠s, one of the shortest paths in G to go from s to v is one of the shortest paths to go from s to the predecessor[v] plus the edge (predecessor[v], v)

# Correctness of BFS

**Proof (contradiction 1/3):** At end of BFS on s, <u>$d[v] = \delta(s,v)$ for all v</u>

1. Assume vertex v with lowest $\delta(s,v)$ such that $d[v] \neq \delta(s,v)$. $v \neq s$
2. $d[v] \geq \delta(s,v)$ *(because of L2)* $\Rightarrow d[v] > \delta(s,v)$
3. v is reachable from s (otherwise, $\delta(s,v) = \infty \geq d[v]$, which contradicts the previous inequality)
4. Let u be the vertex that immediately precedes v in one of the shortest paths : $\delta(s,v)= \delta(s,u)+1$ (therefore $\delta(s,u) < \delta(s,v)$)
5. On the other hand: $d[u]= \delta(s,u)$, because of the way v is chosen

<u>Therefore:</u>

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

# Correctness of BFS

**Proof (contradiction 2/3):**

Now, in the BFS algorithm, after DEQUEUE u from Q

- If v were "white" $\Rightarrow$ d[v]=d[u]+1 (line 15) $\Rightarrow$ **contradiction ***
- If v were "black" v would no longer be in Q, and d[v] $\leq$ d[u] $\Rightarrow$ **contradiction ***
- If v were "gray" $\Rightarrow$ v is gray before DEQUEUE u from Q.
  It was painted gray after removing another vertex w, for which d[v]=d[w]+1

  If w had been removed from Q before u $\Rightarrow$

  $d[w] \leq d[u] \Rightarrow d[v] = d[w]+1 \leq d[u]+1 \Rightarrow$ **contradiction ***

**<u>Therefore d[v]= $\delta$(s,v) for all v</u>**

# Correctness of BFS

**Proof (contradiction 3/3):**

One also concludes that:

- BFS discovers all vertices that are reachable from s. Otherwise, one would have some v for which $\infty = d[v] > \delta(s,v)$, which contradicts the previous result.

- [d[v]=d[u]+1] because [predecessor[v]=u]. Therefore, one of the shortest paths to go from s to v is the shortest path to go from s to predecessor[v] plus the edge (predecessor[v], v)

# Proofs of the properties of BFS

1. BFS finds the shortest path between two nodes

2. BFS is correct

3. **BFS finds the BF tree**

# Breadth-first tree

Def. Given $G=(V,E)$ and $s$, we define $G_p=(V_p, E_p)$, the predecessor subgraph of $G$, where :

$$V_p=\{v \in V : p[v] \neq \text{NIL}\} \cup \{s\}$$

$$E_p=\{(p(v), v): v \in V_p - \{s\}\}$$

$G_p$ is a **breadth-first tree** if:

- $V_p$ = all vertices $v$ that are reachable from $s$
- There is a single path from $s$ to $v$ in $G_p$, which is the shortest path from $s$ to $v$ in $G$.

# Breadth-first tree

**<u>L:</u>** BFS builds a data structure with **predecessor** such that the corresponding predecessor subgraph $G_p=(V_p, E_p)$ is a **breadth-first tree**

**Proof:**

# Breadth-first tree

**L:** BFS builds a data structure with **predecessor** such that the corresponding predecessor subgraph $G_p=(V_p, E_p)$ is a **breadth-first tree**

**Proof:**

p[v]=u (line 16), iff $(u,v)\in E$ and $\delta(s,v) < \infty$ (that is, if v is reachable from s)

$\Rightarrow V_p$ contains all vertices that are reachable from s

The tree $G_p$ has a single path for each $v\in V_p \Rightarrow$

All these paths are shortest paths (previous theorem)