

CS 344: Design and Analysis of Computer Algorithms

(Spring 2022 — Sections 5,6,7,8)

Lecture 16: Graph Search and Shortest Paths: BFS

Graph Search Algorithm 2: Breadth-First-Search (BFS)

BFS

- Another natural strategy for exploring the graph:
- Mark **s**; then all neighbors of **s**; then neighbors of neighbors of **s**, and continue like this...

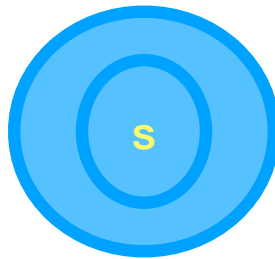
BFS

- Another natural strategy for exploring the graph:
- Mark **s**; then all neighbors of **s**; then neighbors of neighbors of **s**, and continue like this...



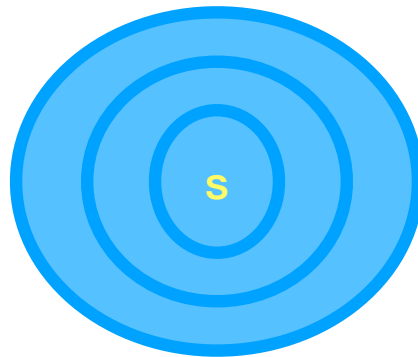
BFS

- Another natural strategy for exploring the graph:
- Mark **s**; then all neighbors of **s**; then neighbors of neighbors of **s**, and continue like this...



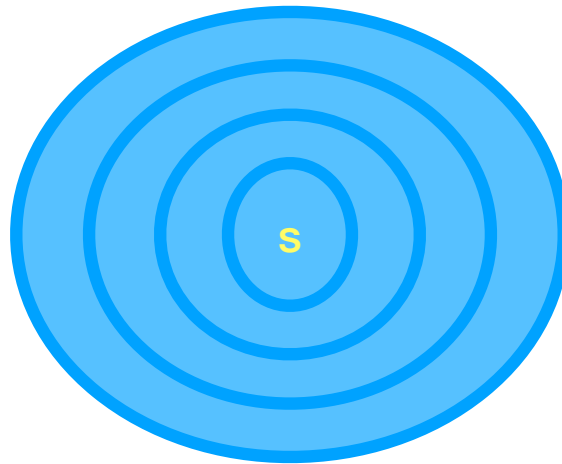
BFS

- Another natural strategy for exploring the graph:
- Mark **s**; then all neighbors of **s**; then neighbors of neighbors of **s**, and continue like this...



BFS

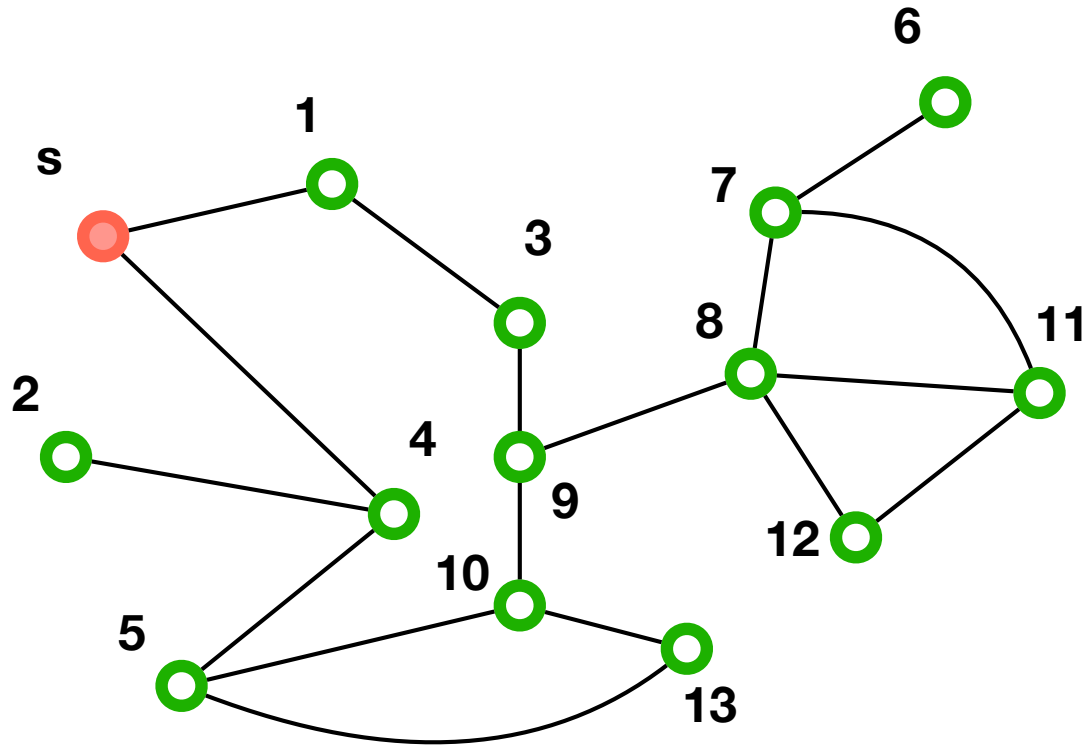
- Another natural strategy for exploring the graph:
- Mark **s**; then all neighbors of **s**; then neighbors of neighbors of **s**, and continue like this...







BFS

- Initialize $\text{mark}[1:n] = \text{'false'}$ for every vertex v
- Create a queue data structure Q and insert s to Q
- While Q is not empty;
 - Remove v from the top of the queue
 - If $\text{mark}[v] = \text{'true'}$ continue to the next iteration of the while-loop
 - Otherwise, $\text{mark}[v] = \text{'true'}$; add all neighbors $N(v)$ to queue Q
- Return all vertices that are marked (true).

BFS: Example

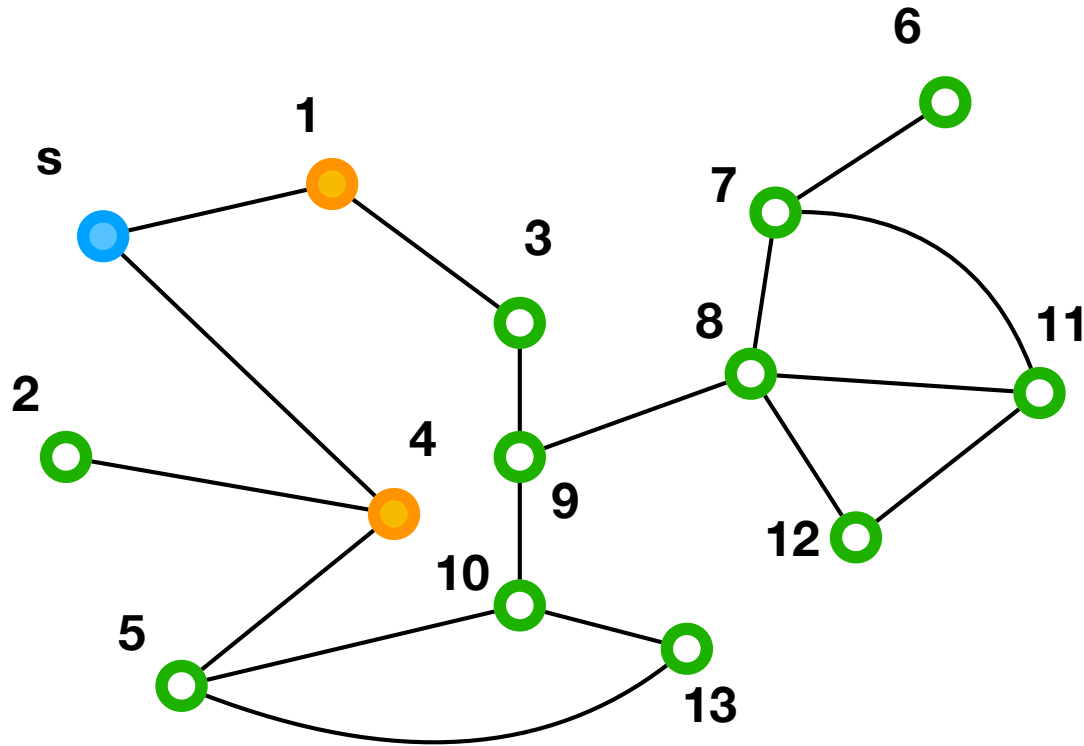






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

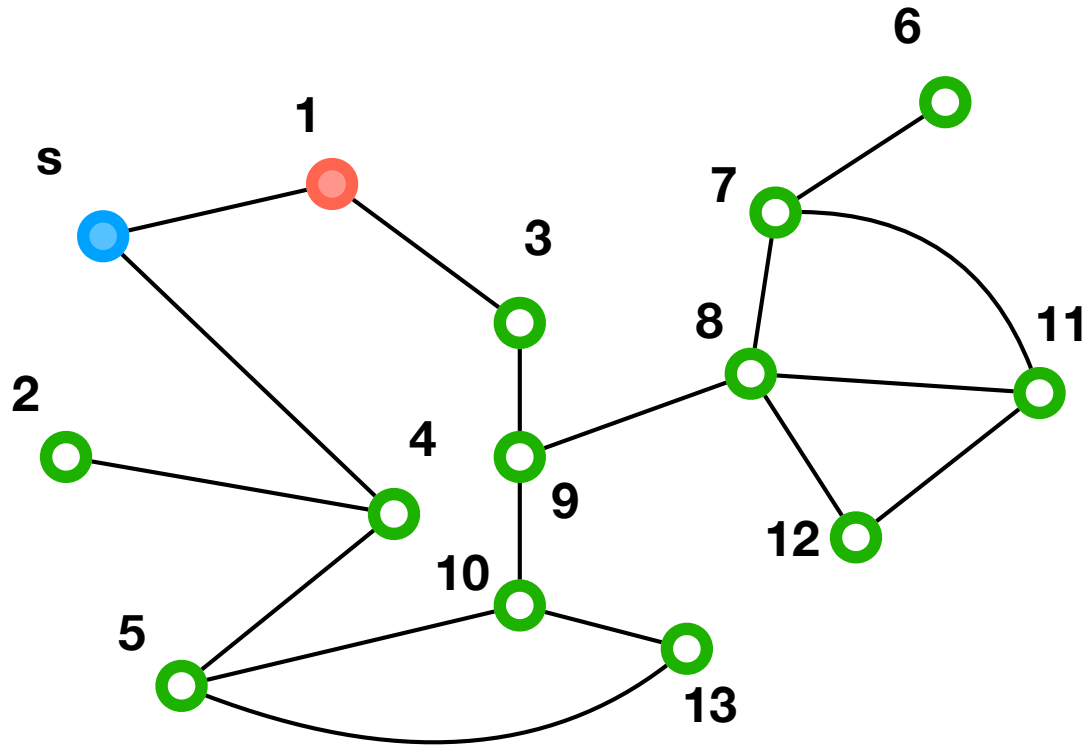






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

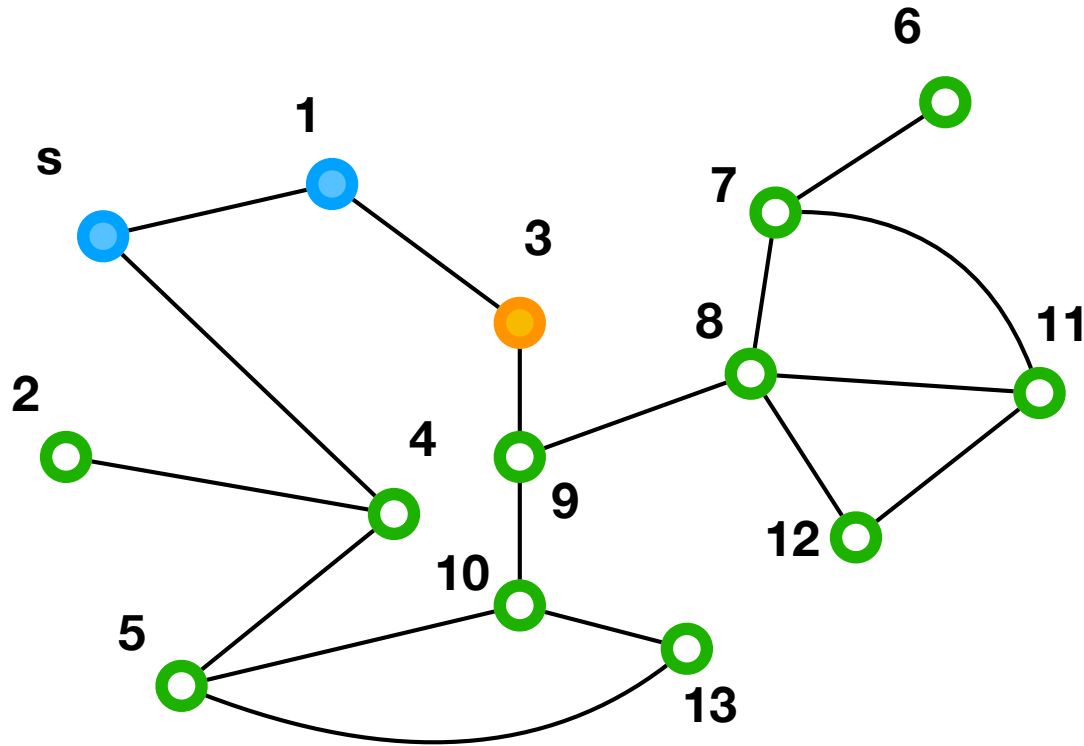






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

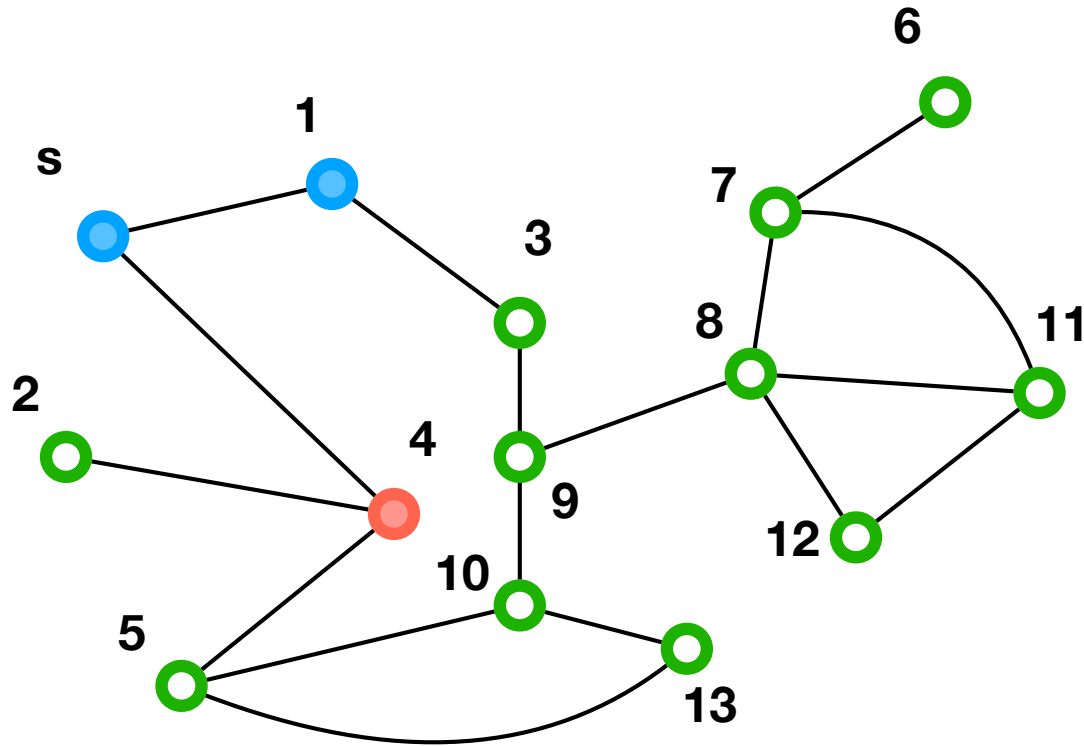






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

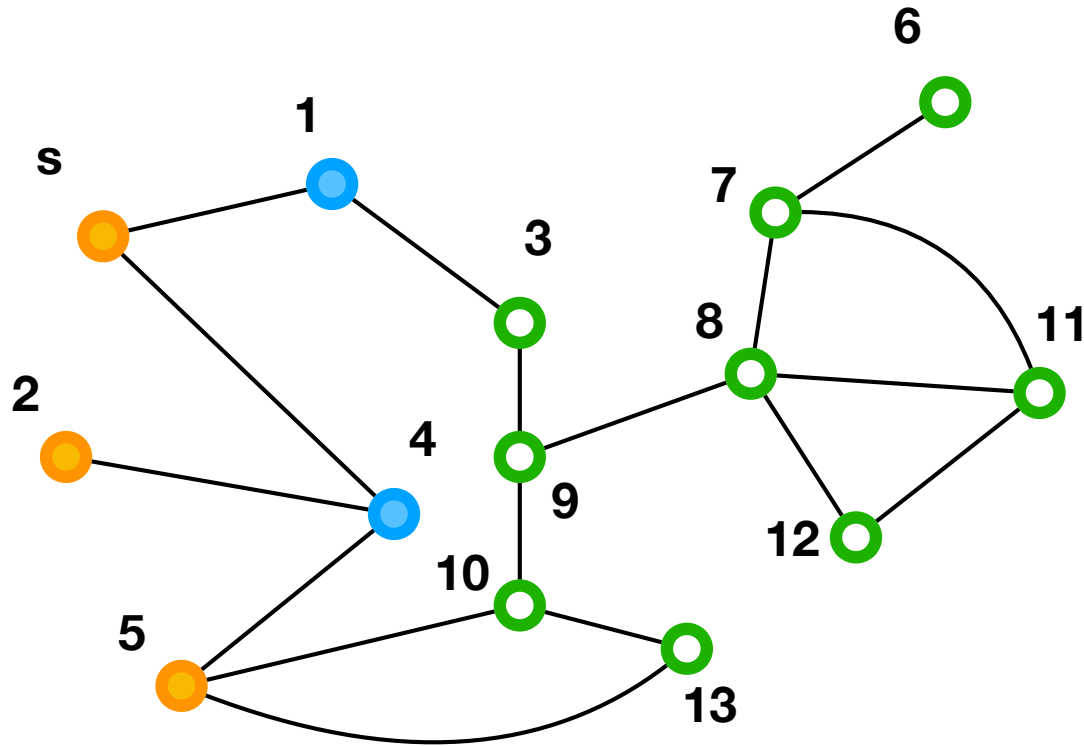






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

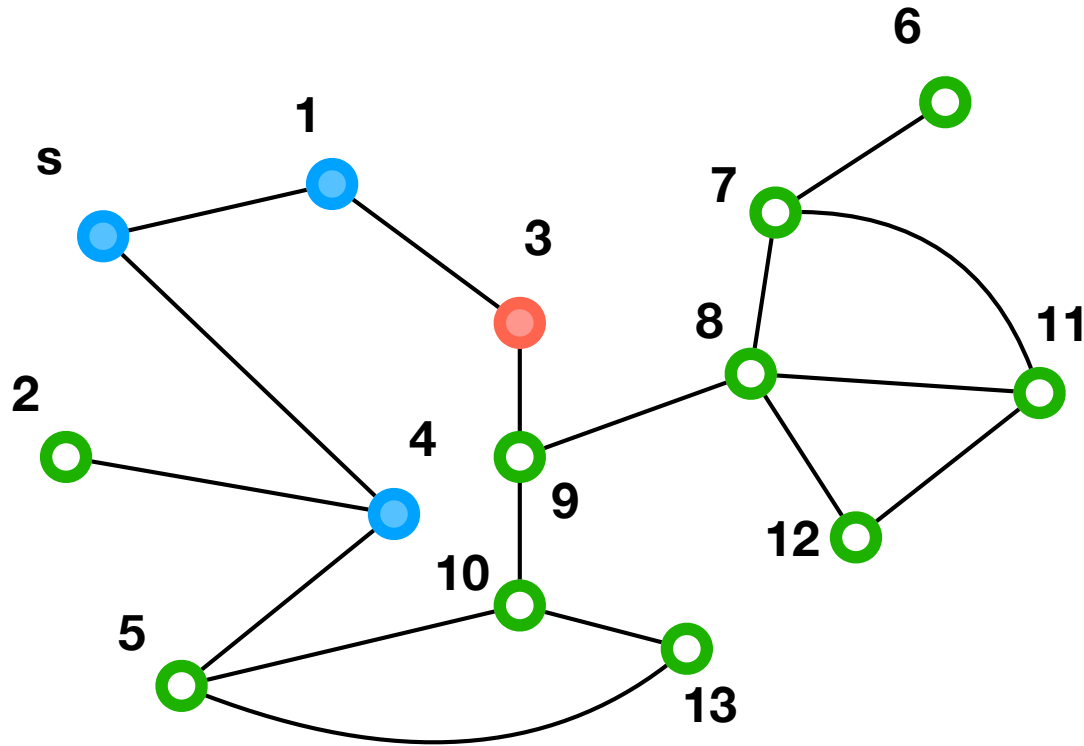






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

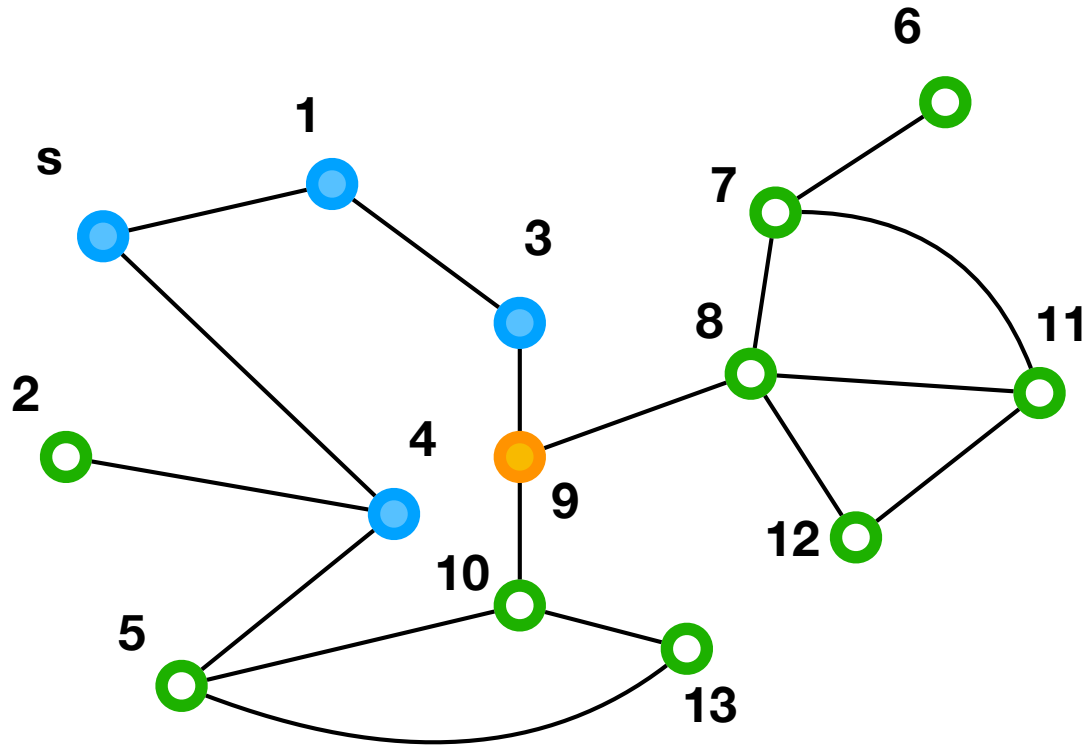






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

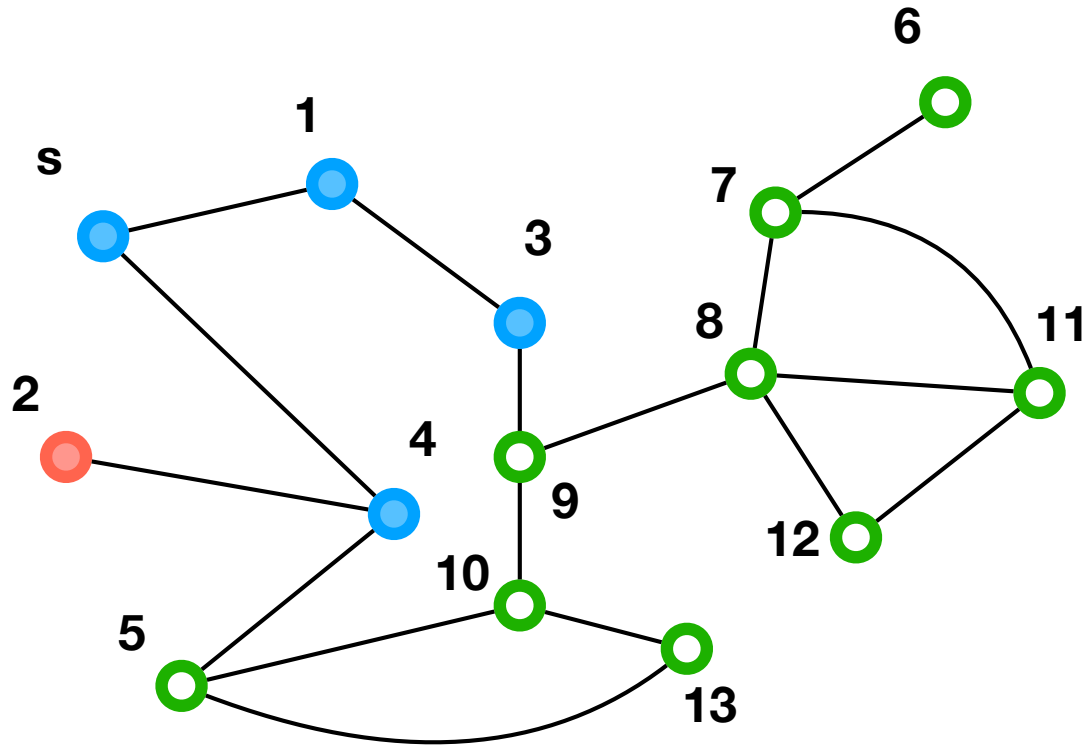






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

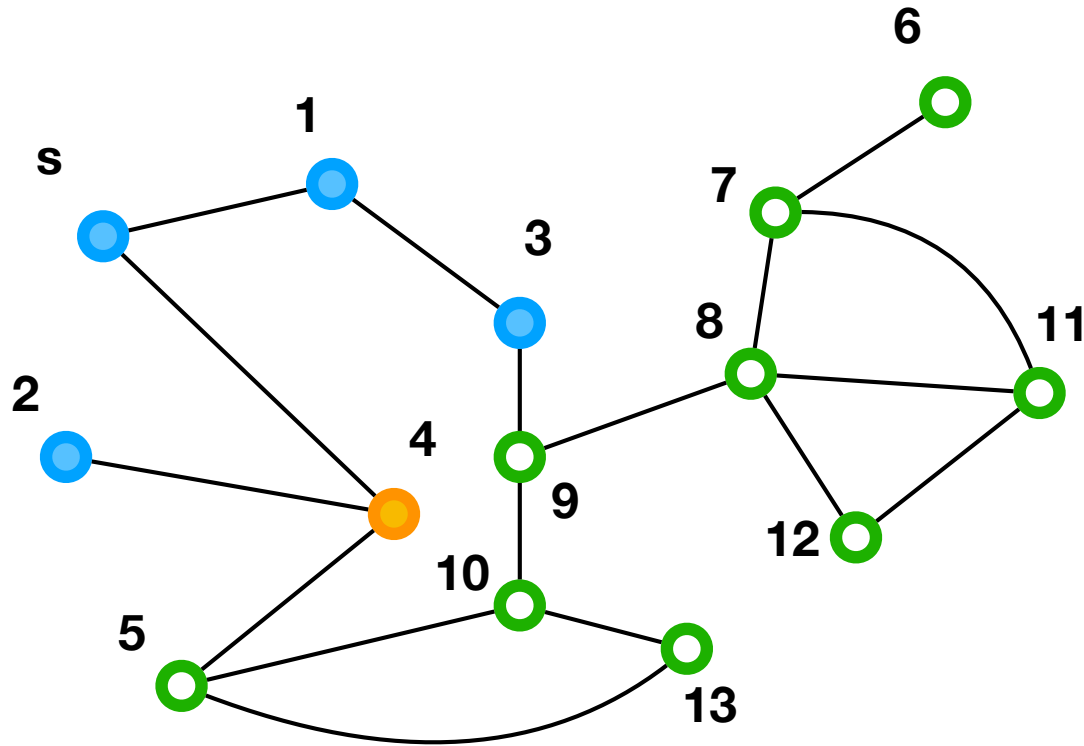






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

2	5	s	9					
---	---	---	---	--	--	--	--	--

queue Q

BFS: Example

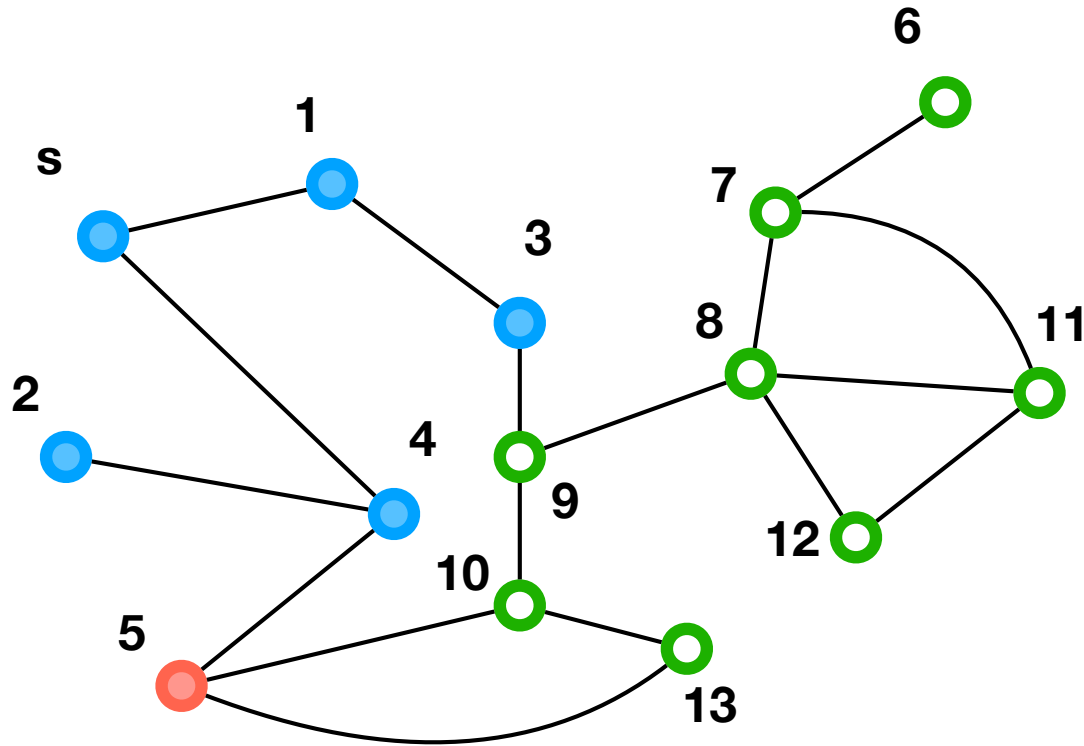






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

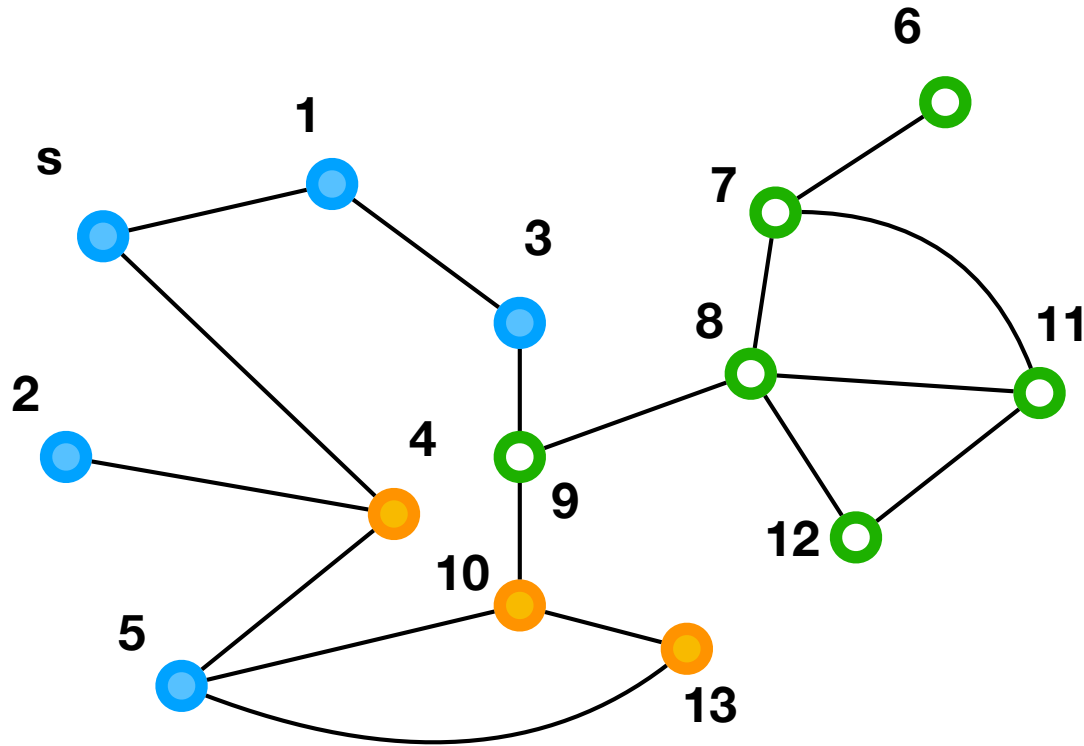






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q



queue Q

BFS: Example

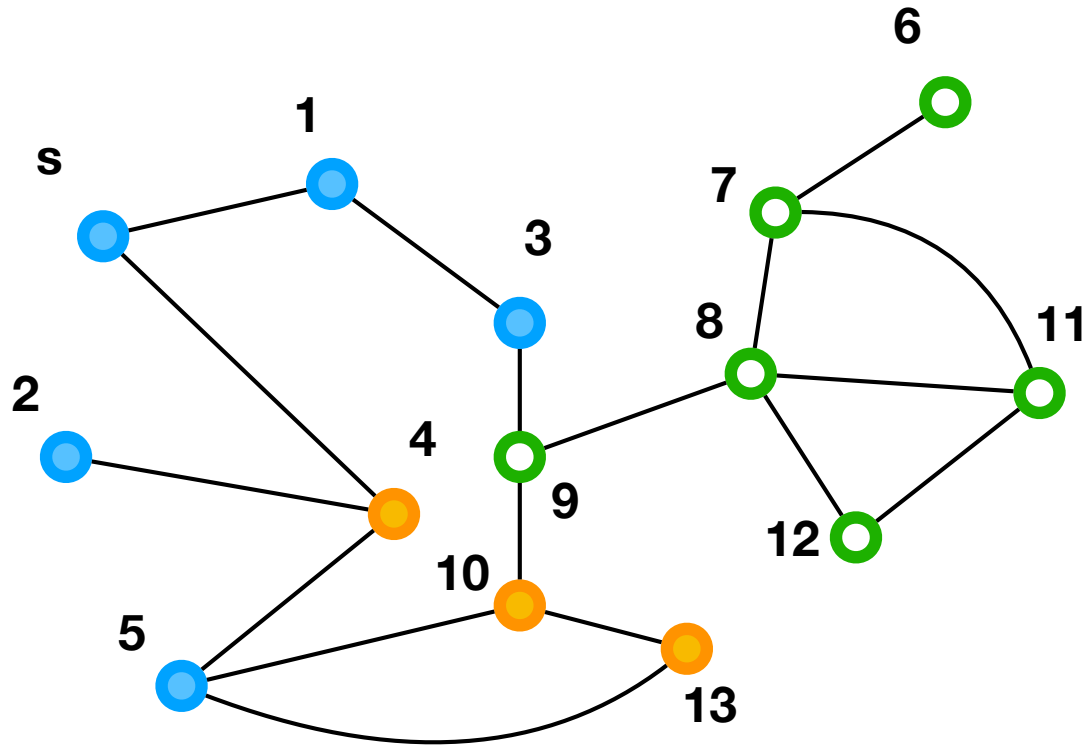






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	9	4	4	10	13			
---	---	---	---	----	----	--	--	--

queue Q

BFS: Example

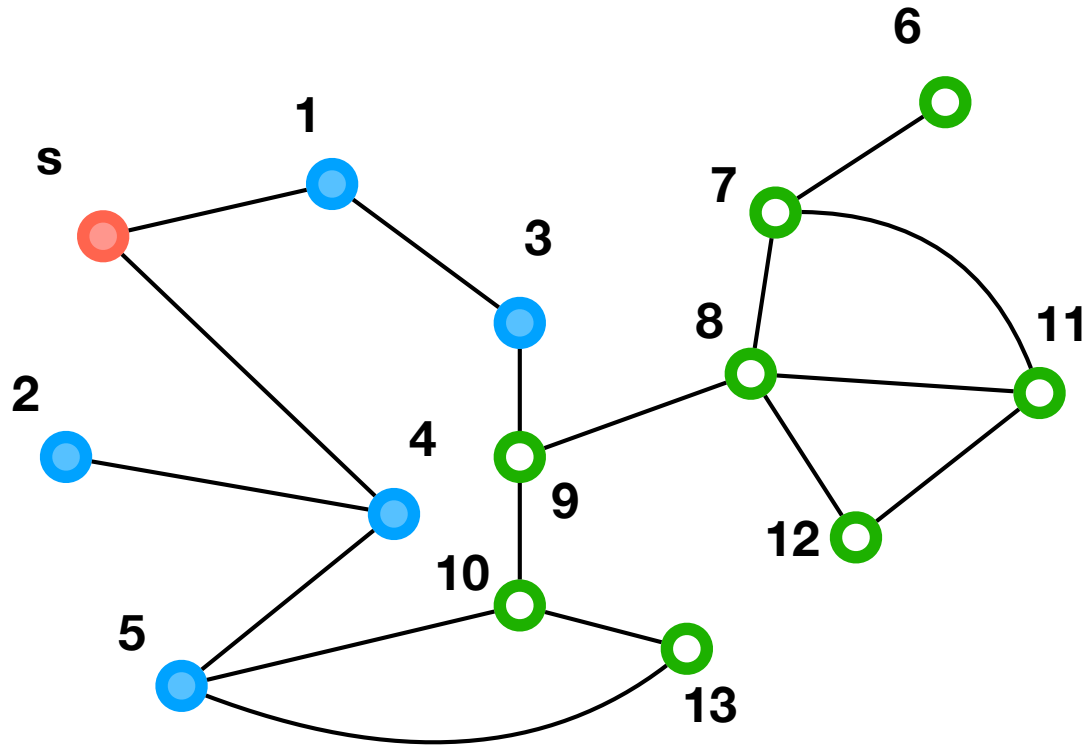






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	9	4	4	10	13			
---	---	---	---	----	----	--	--	--

queue Q

BFS: Example

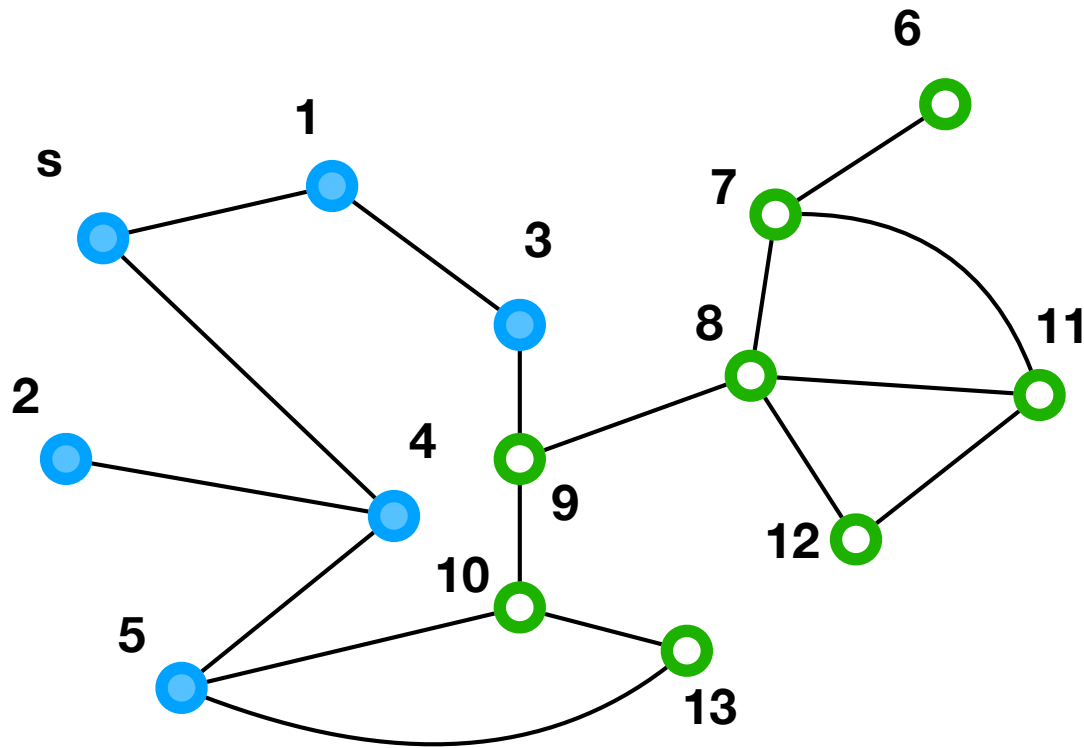






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	9	4	4	10	13			
---	---	---	---	----	----	--	--	--

queue Q

BFS: Example

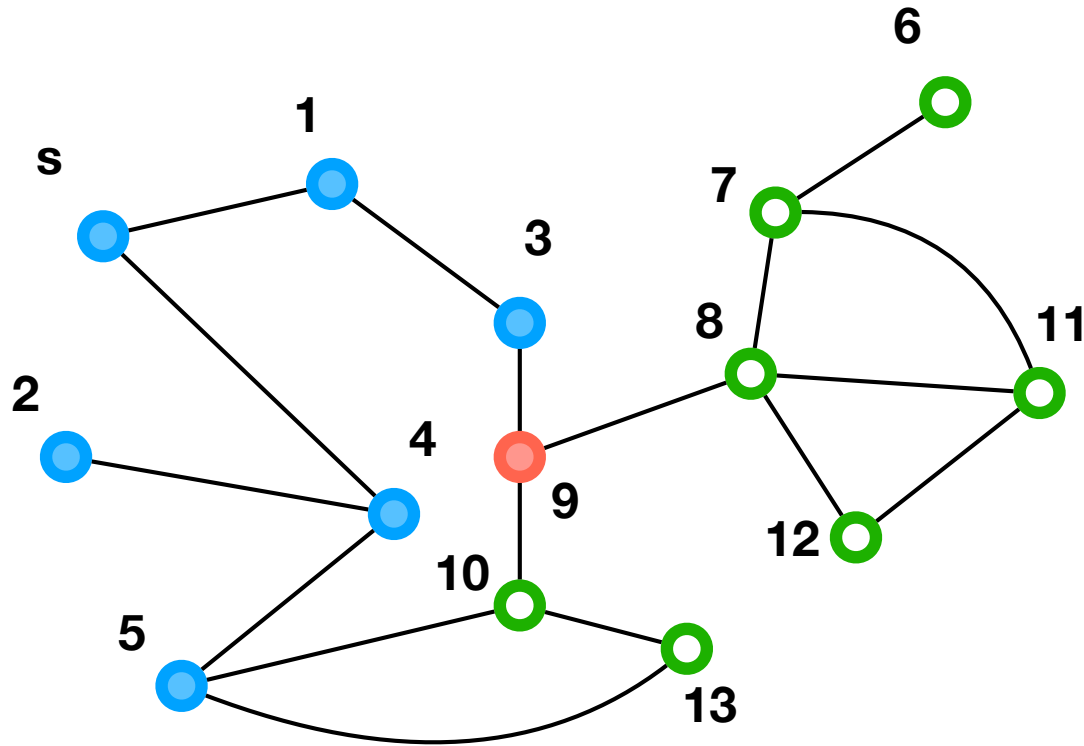






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

9	4	4	10	13				
---	---	---	----	----	--	--	--	--

queue Q

BFS: Example

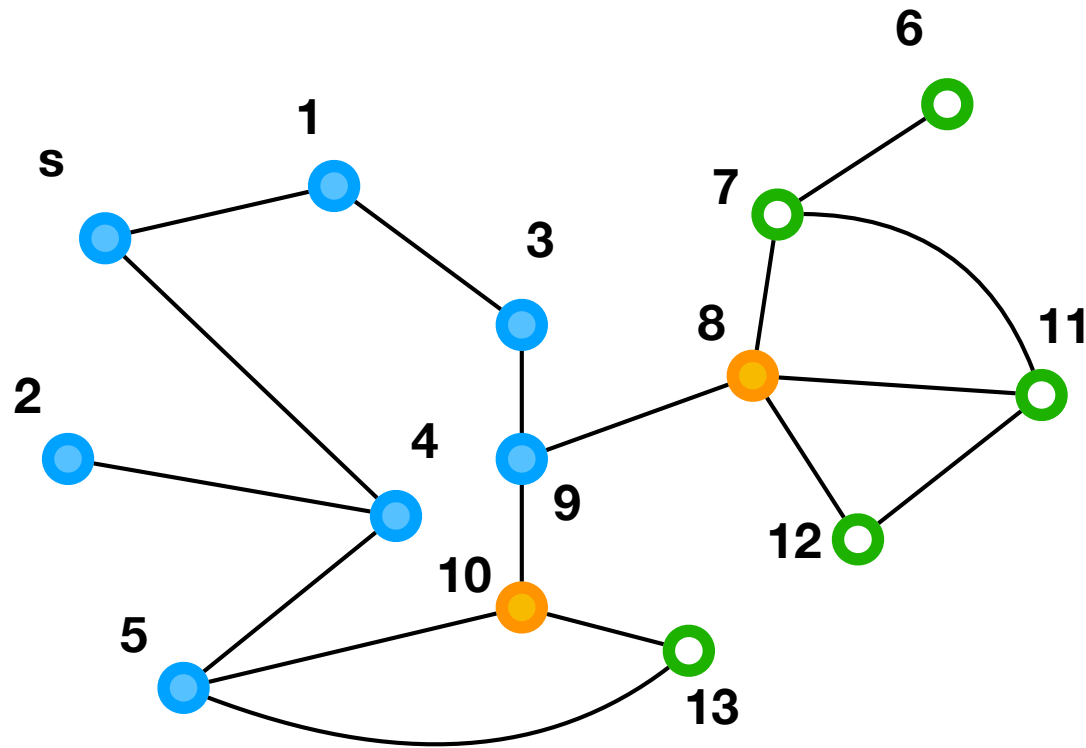






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

9	4	4	10	13				
---	---	---	----	----	--	--	--	--

queue Q

BFS: Example

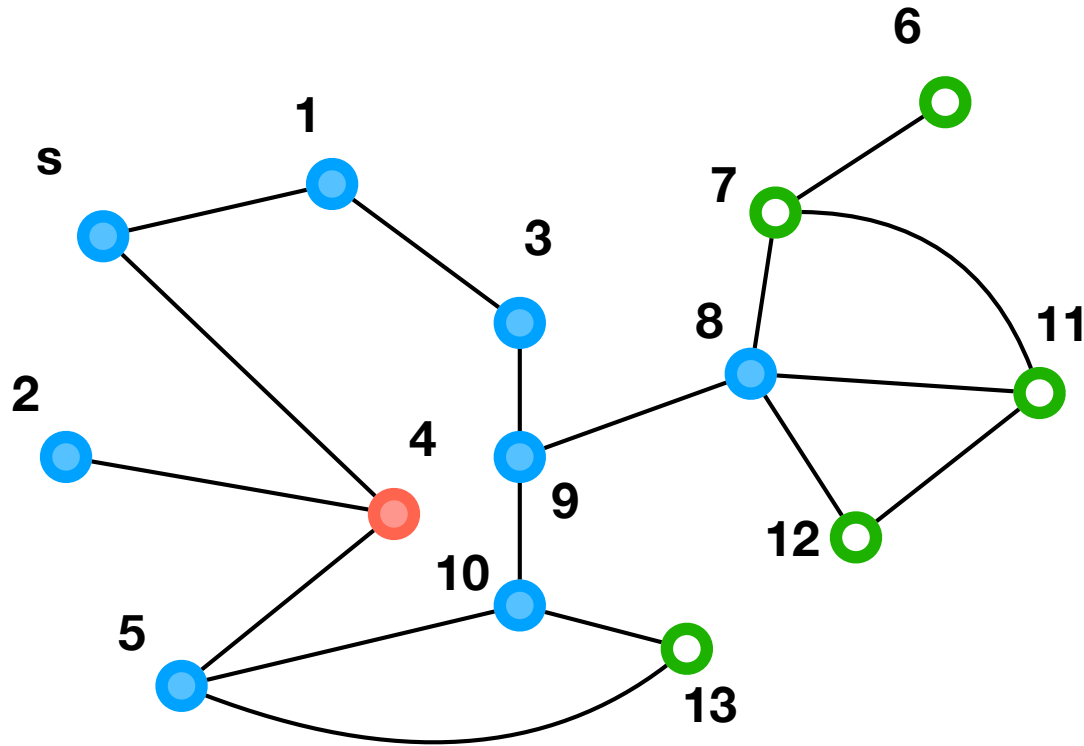


-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

4	4	10	13	8	10			
---	---	----	----	---	----	--	--	--

queue Q

BFS: Example

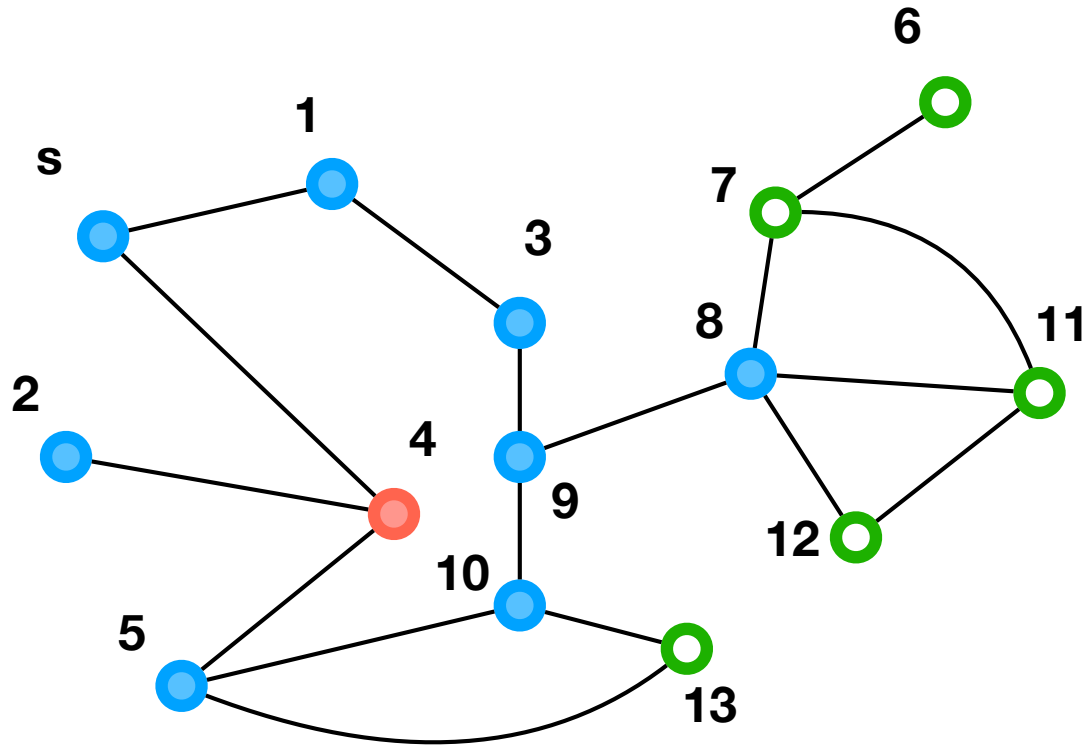


- marked vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

4	4	10	13	8	10			
---	---	----	----	---	----	--	--	--

queue Q

BFS: Example

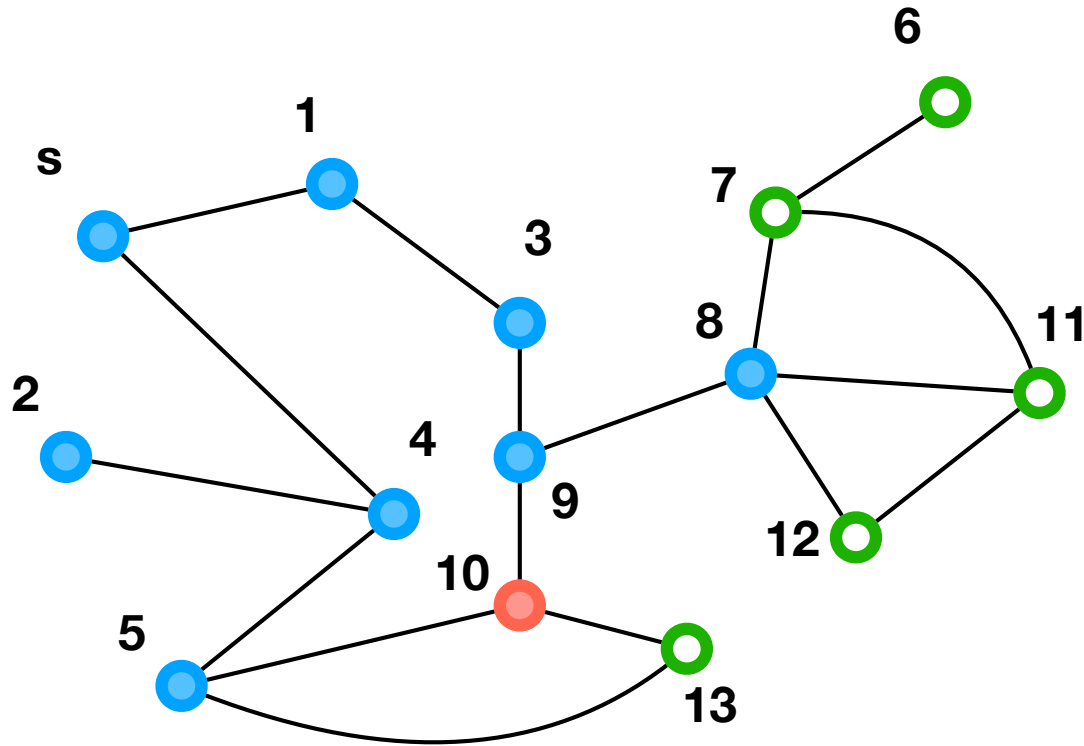






- marked vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

4	10	13	8	10				
---	----	----	---	----	--	--	--	--

queue Q

BFS: Example

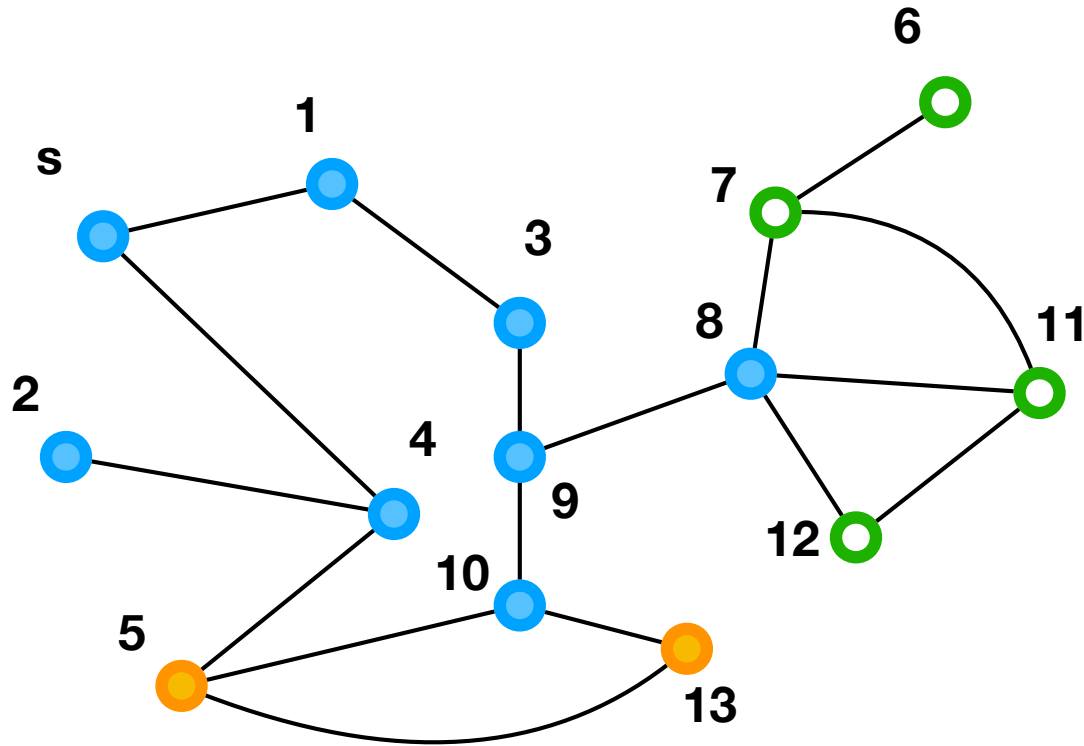






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

10	13	8	10					
----	----	---	----	--	--	--	--	--

queue Q

BFS: Example

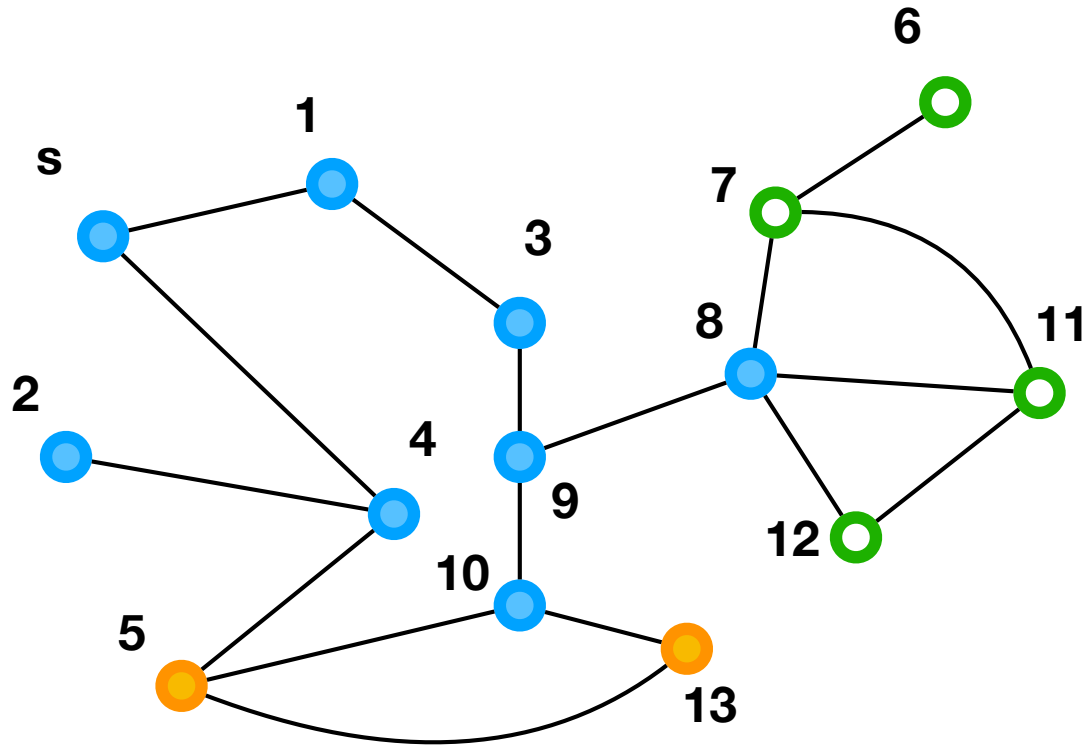






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

13	8	10	5	13			
----	---	----	---	----	--	--	--

queue Q

BFS: Example

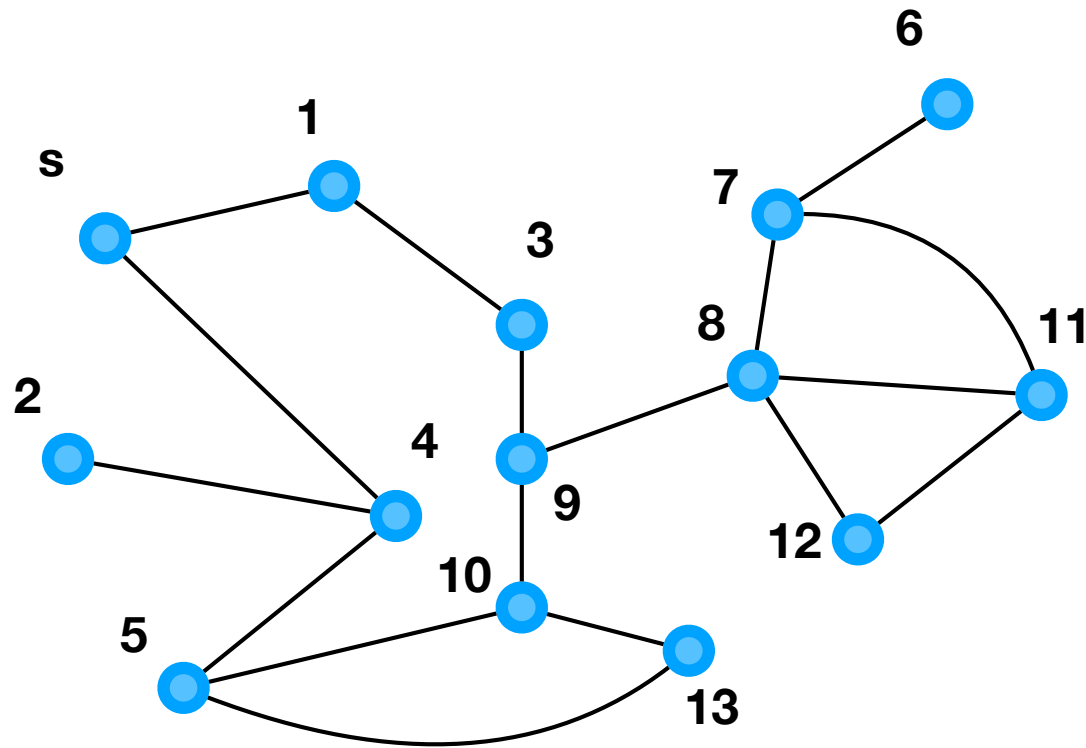






-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

13	8	10	5	13			
----	---	----	---	----	--	--	--

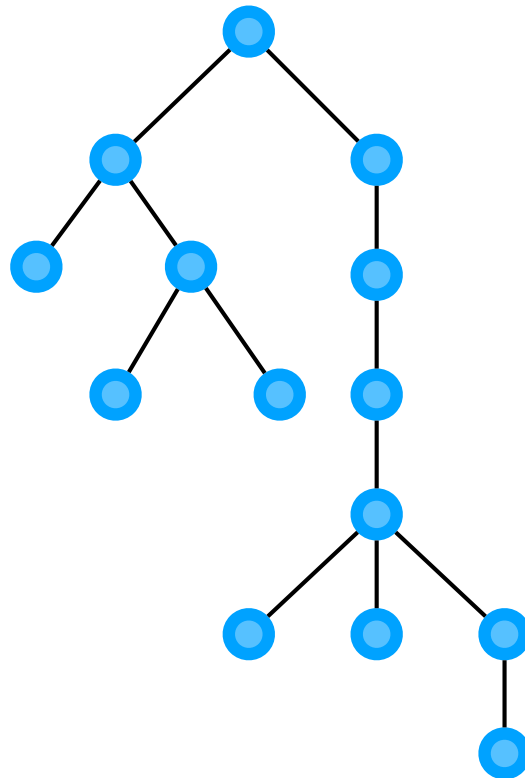
queue Q

BFS: Example

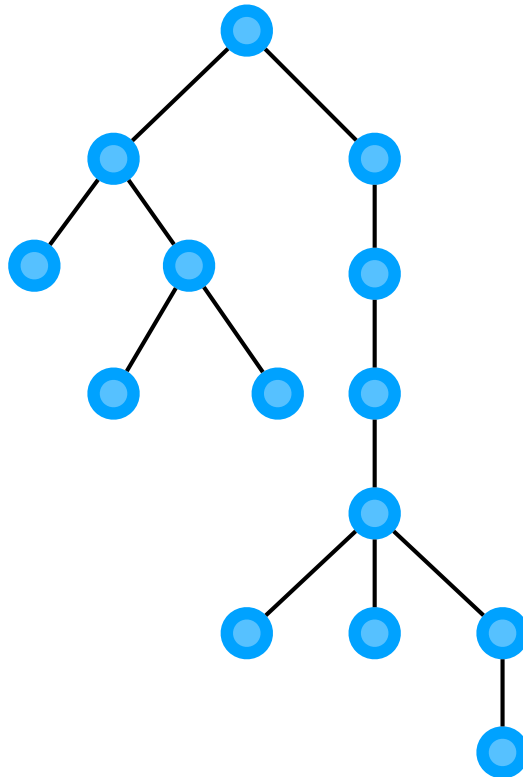


-  marked vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

BFS Tree



BFS Tree



BFS tends to create **very shallow** trees

BFS

- Initialize $\text{mark}[1:n] = \text{'false'}$ for every vertex v
- Create a queue data structure Q and insert s to Q
- While Q is not empty;
 - Remove v from the top of the queue
 - If $\text{mark}[v] = \text{'true'}$ continue to the next iteration of the while-loop
 - Otherwise, $\text{mark}[v] = \text{'true'}$; add all neighbors $N(v)$ to queue Q
- Return all vertices that are marked (true).

Proof of Correctness

- Postponed for now — we will prove a generalization soon

Runtime Analysis

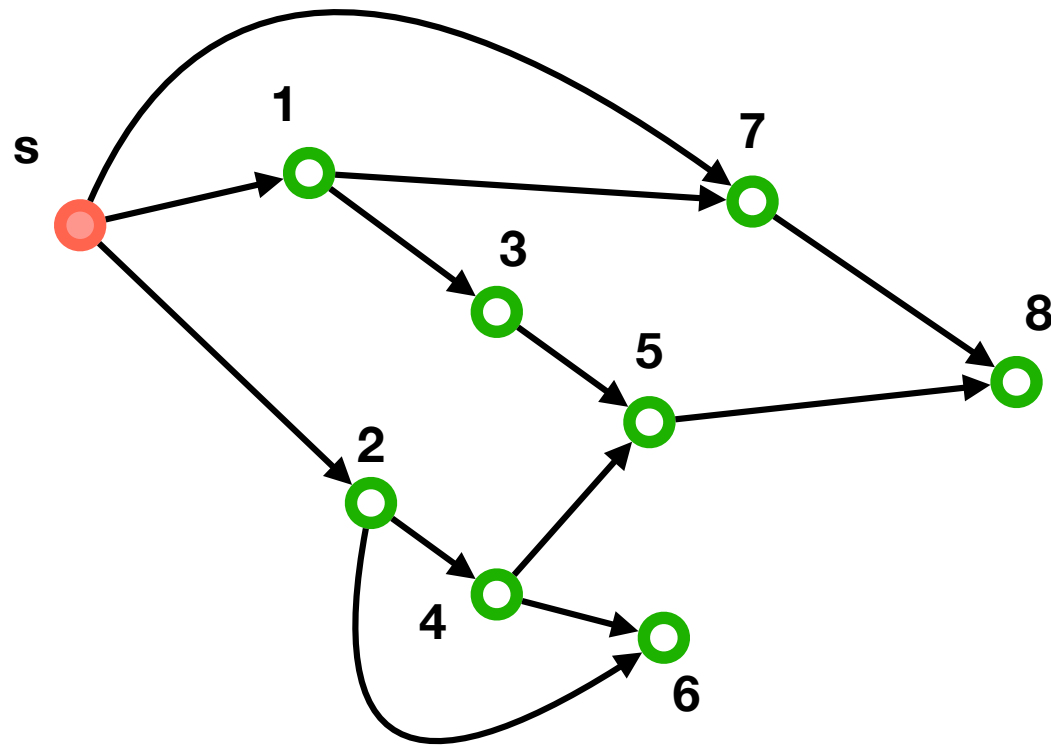
- Initialize $\text{mark}[1:n] = \text{'false'}$
- Create a queue data structure Q and insert s to Q
- While Q is not empty;
 - Remove v from the top of the queue
 - If $\text{mark}[v] = \text{'true'}$ continue
 - Otherwise, $\text{mark}[v] = \text{'true'}$; add all neighbors $N(v)$ to queue Q
- Return all vertices that are marked (true).
- Any vertex is only inserted to the Q once effectively
- Any other time it is inserted, ignore its time and instead “charge” the time to the vertex that inserted it
- The time spent for vertex v is then $O(1 + \text{deg}(v))$
- Total runtime $O(n+m)$ as before

Unweighted Shortest Path

BFS finds Shortest Paths

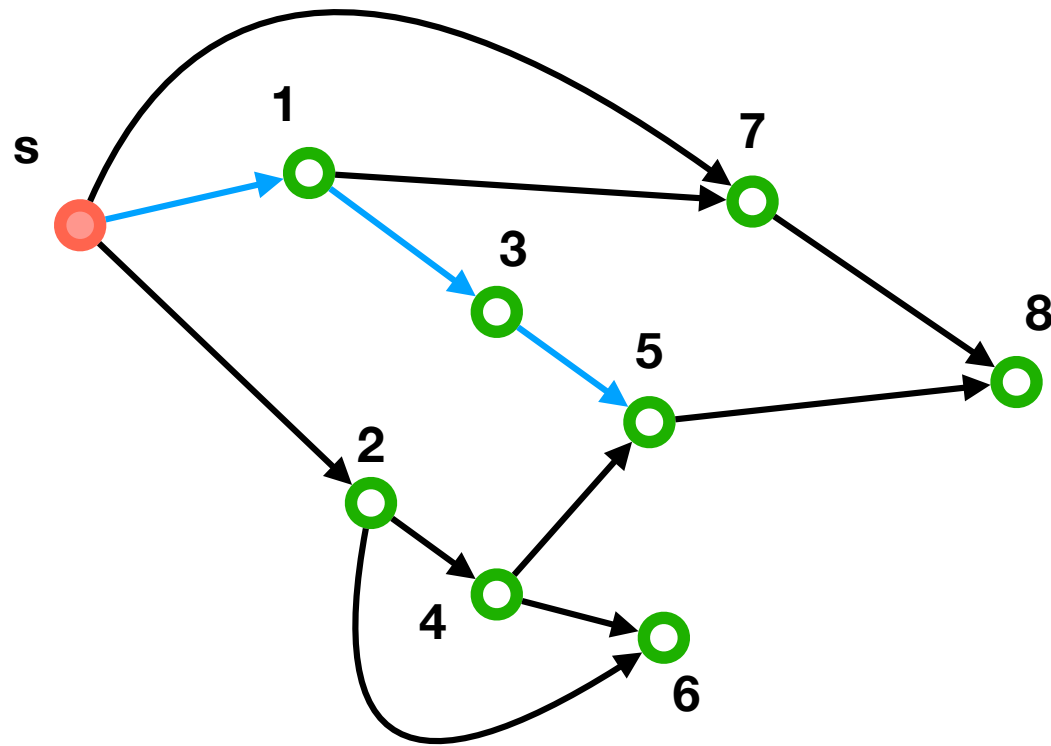
- For any vertex $v \in V$:
 - $\text{dist}(s,v)$: distance of v from s
 - length of shortest path from s to v .

BFS: Example



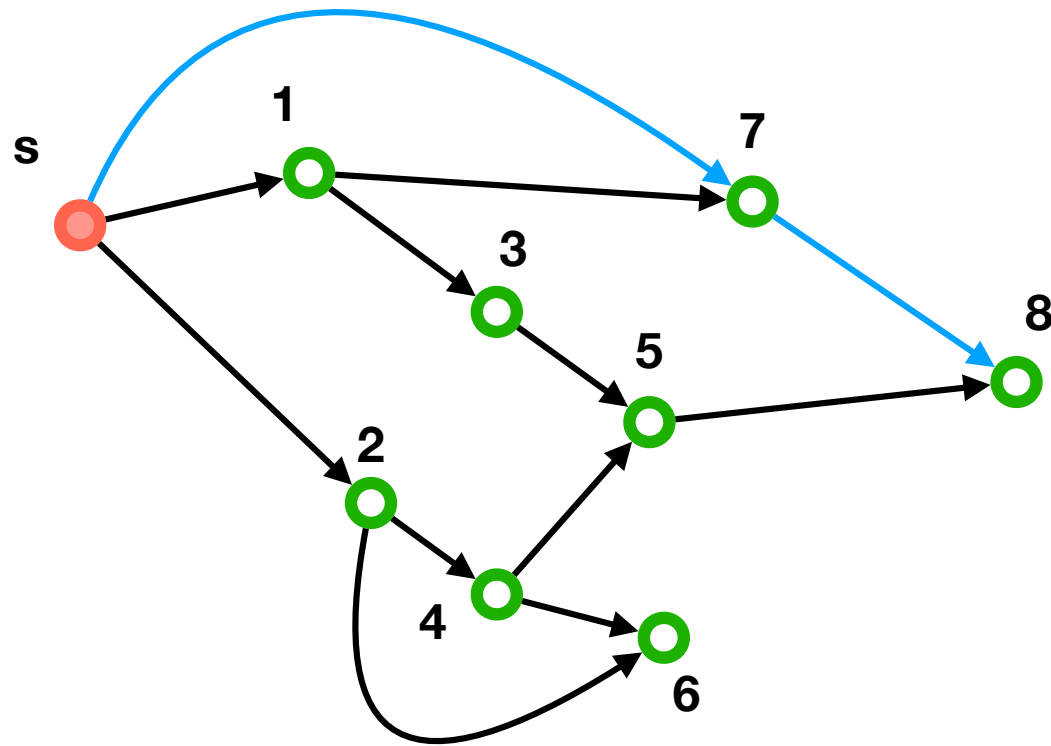
s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2

BFS: Example



s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2

BFS: Example

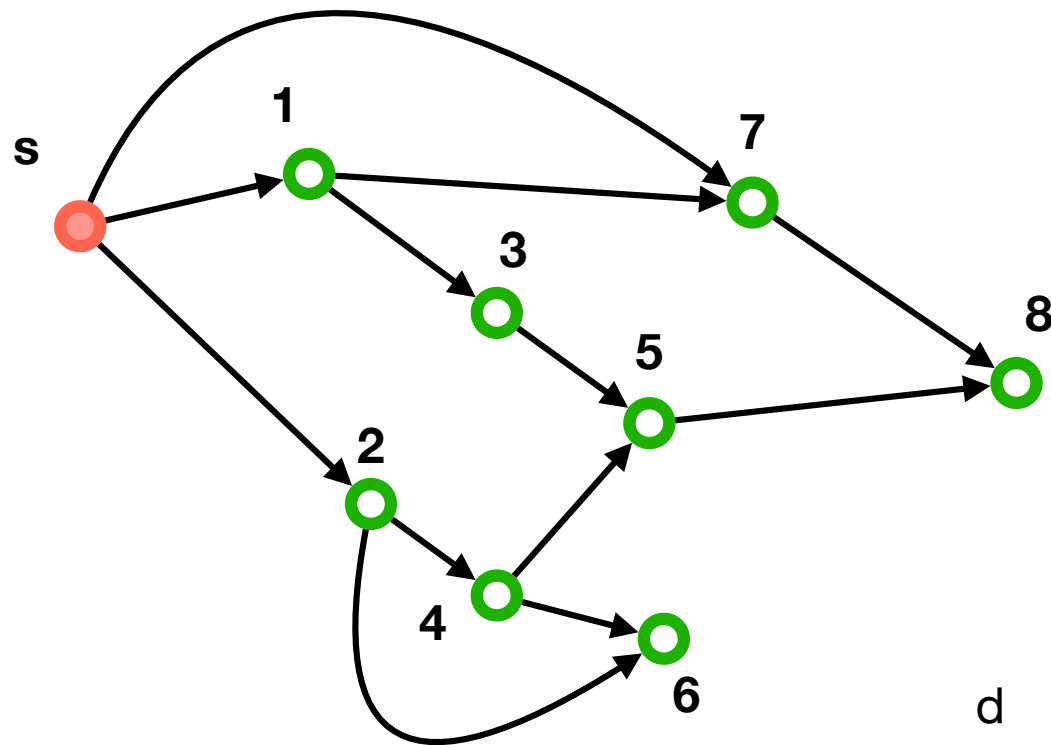


s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2

Shortest-Path-BFS

- Initialize $d[1:n] = +\infty$ for every vertex v
- Create a queue data structure Q and insert s to Q ; set $d[s] = 0$
- While Q is not empty;
 - Remove u from the top of the queue
 - For any $v \in N(u)$:
 - If $d[v] \neq +\infty$ continue to the next neighbor
 - Otherwise, let $d[v] = d[u] + 1$ and add v to Q

Shortest-Path-BFS: Example

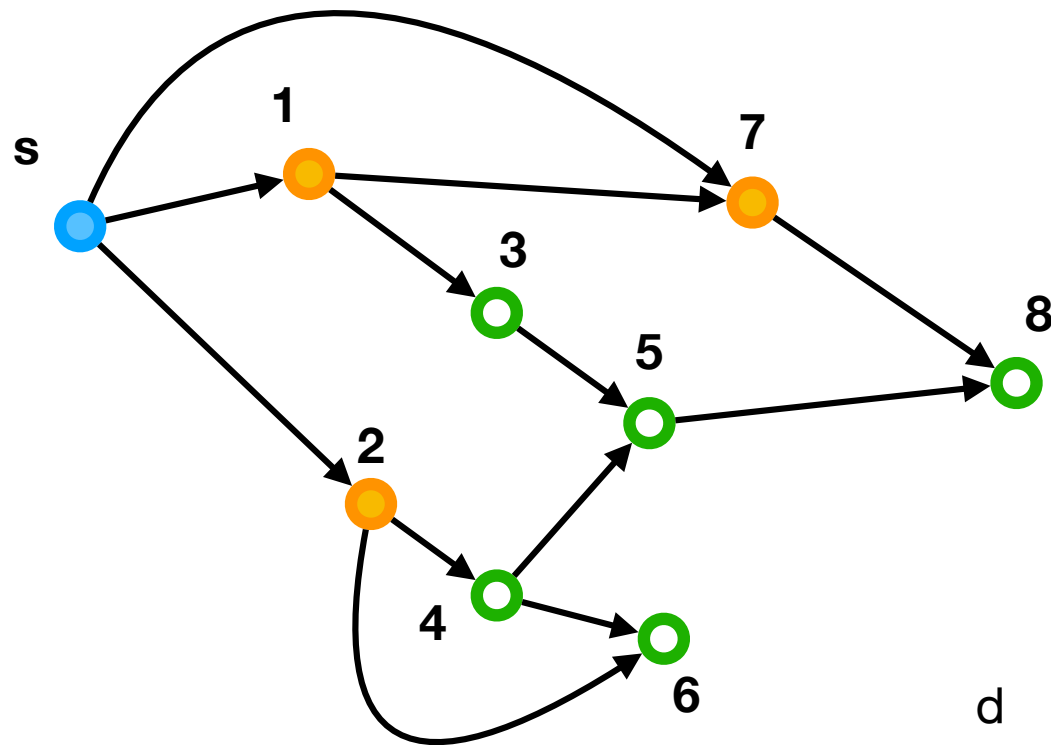


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
s								

queue Q

Shortest-Path-BFS: Example

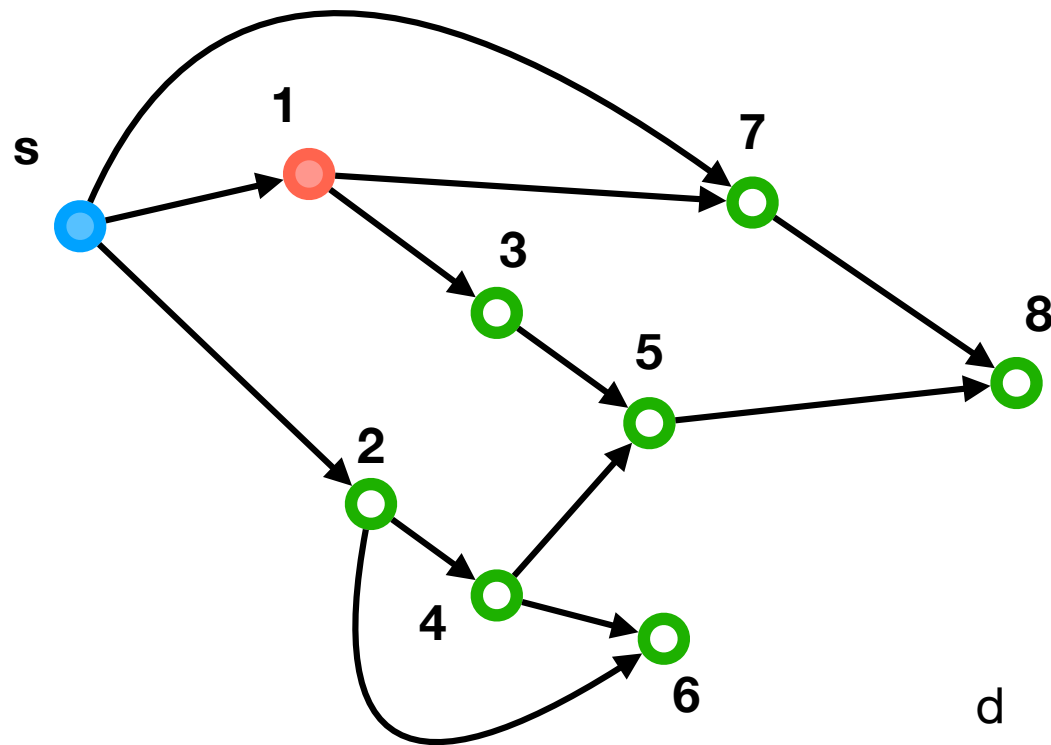


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	$+\infty$	$+\infty$	$+\infty$	$+\infty$	1	$+\infty$
1	2	7						

queue Q

Shortest-Path-BFS: Example

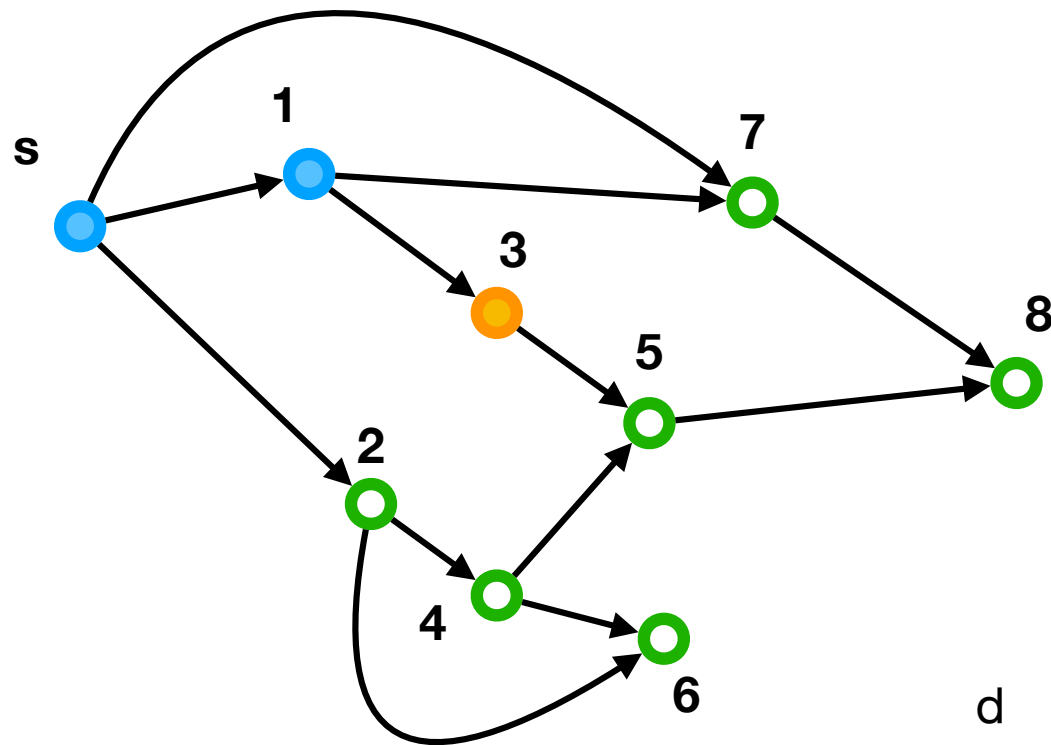


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	$+\infty$	$+\infty$	$+\infty$	$+\infty$	1	$+\infty$
1	2	7						

queue Q

Shortest-Path-BFS: Example

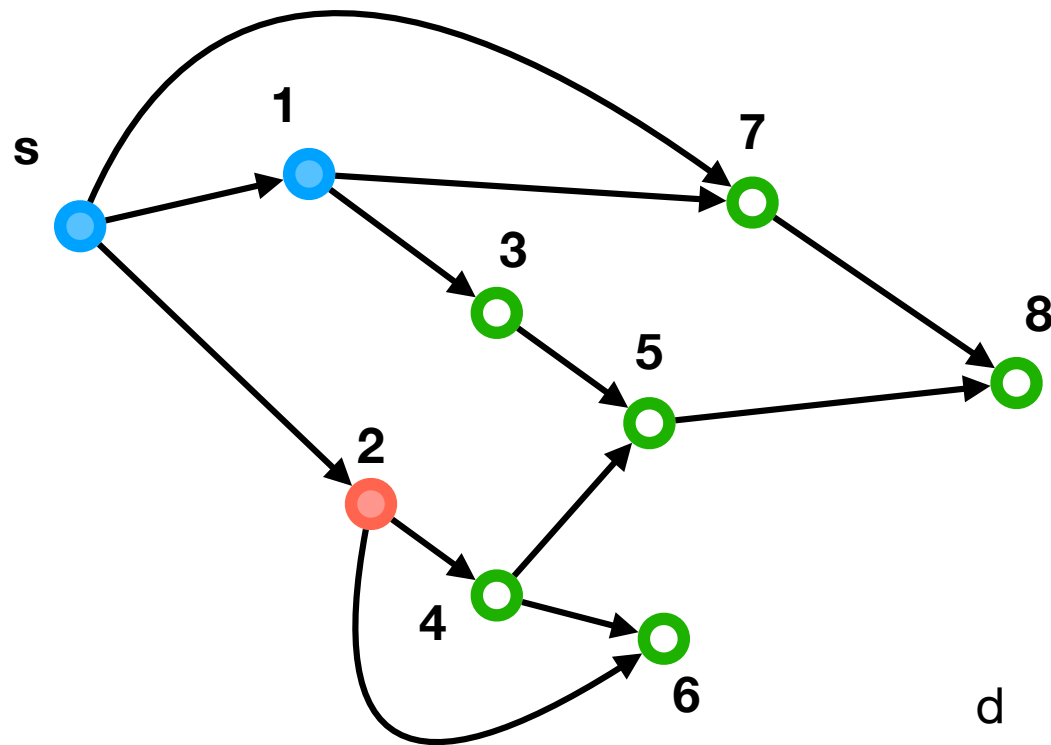


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	$+\infty$	$+\infty$	$+\infty$	1	$+\infty$
2	7	3						

queue Q

Shortest-Path-BFS: Example

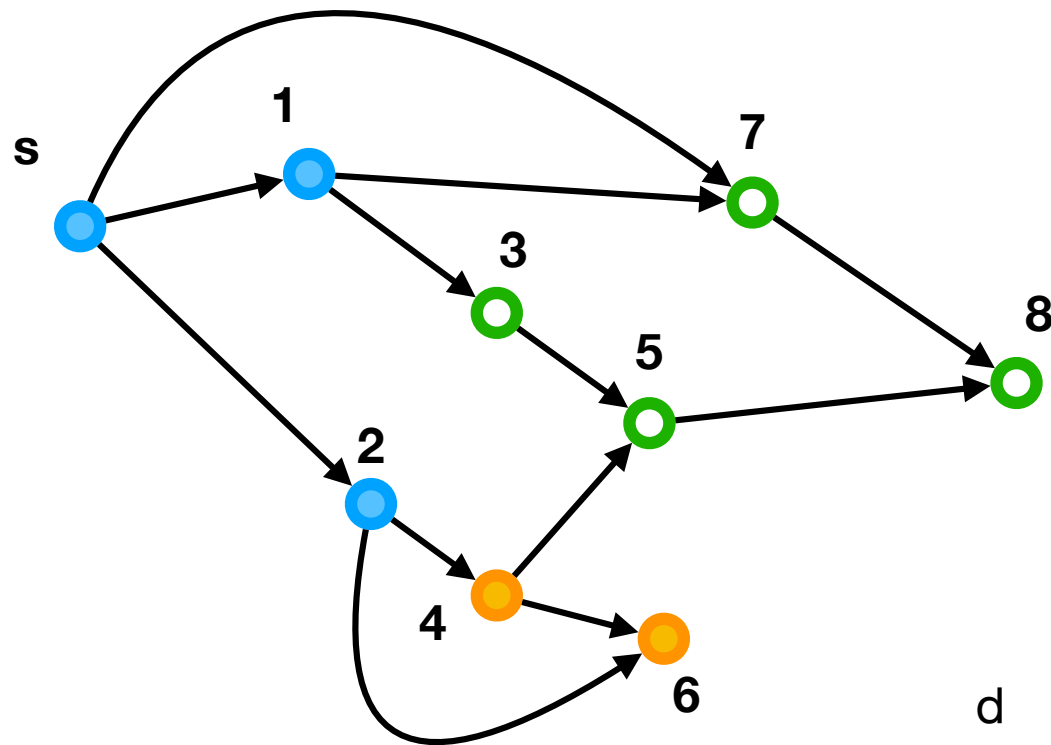


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	$+\infty$	$+\infty$	$+\infty$	1	$+\infty$
2	7	3						

queue Q

Shortest-Path-BFS: Example

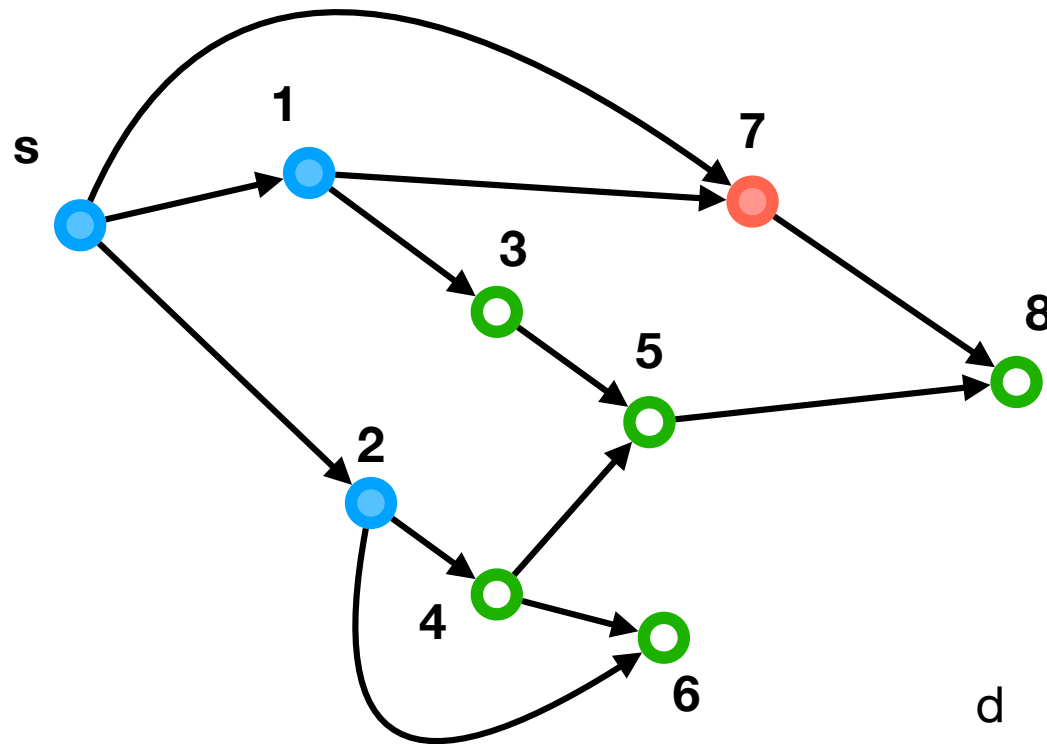


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	$+\infty$	2	1	$+\infty$
7	3	4	6					

queue Q

Shortest-Path-BFS: Example

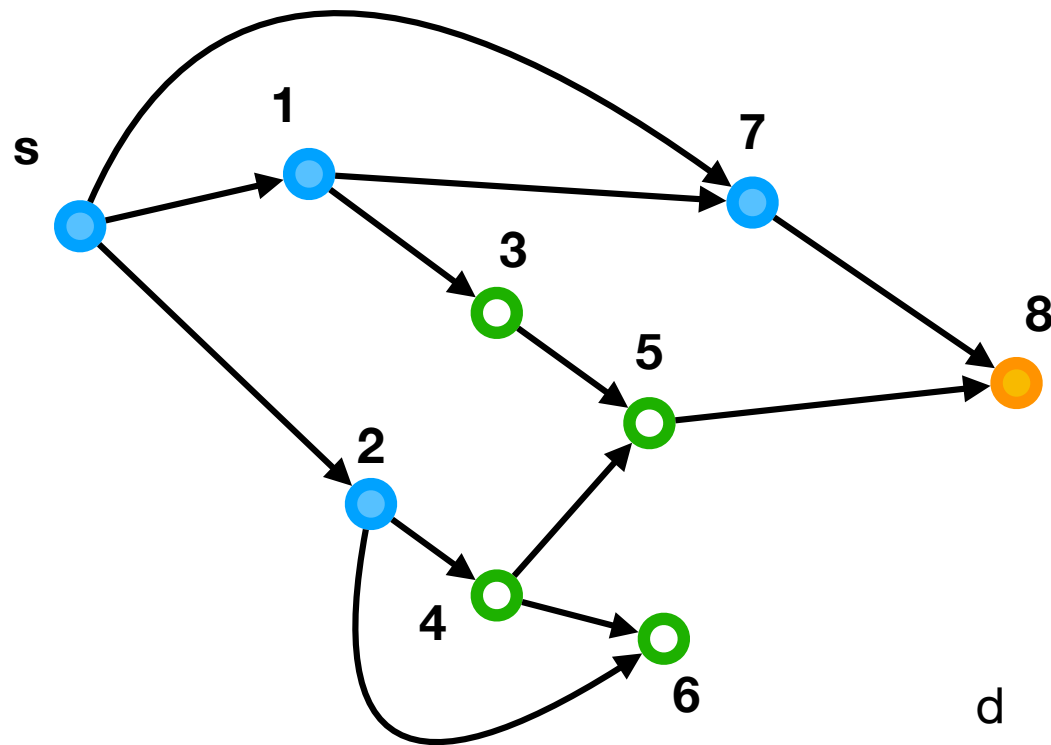


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	$+\infty$	2	1	$+\infty$
7	3	4	6					

queue Q

Shortest-Path-BFS: Example

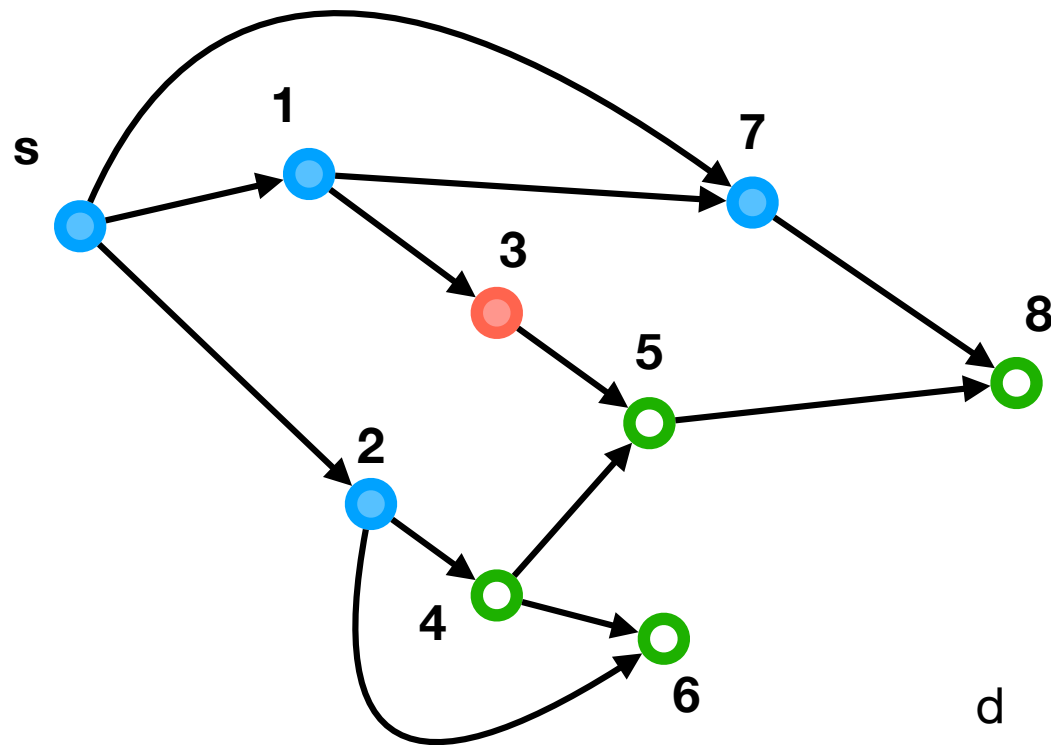


- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	$+\infty$	2	1	2
3	4	6	8					

queue Q

Shortest-Path-BFS: Example

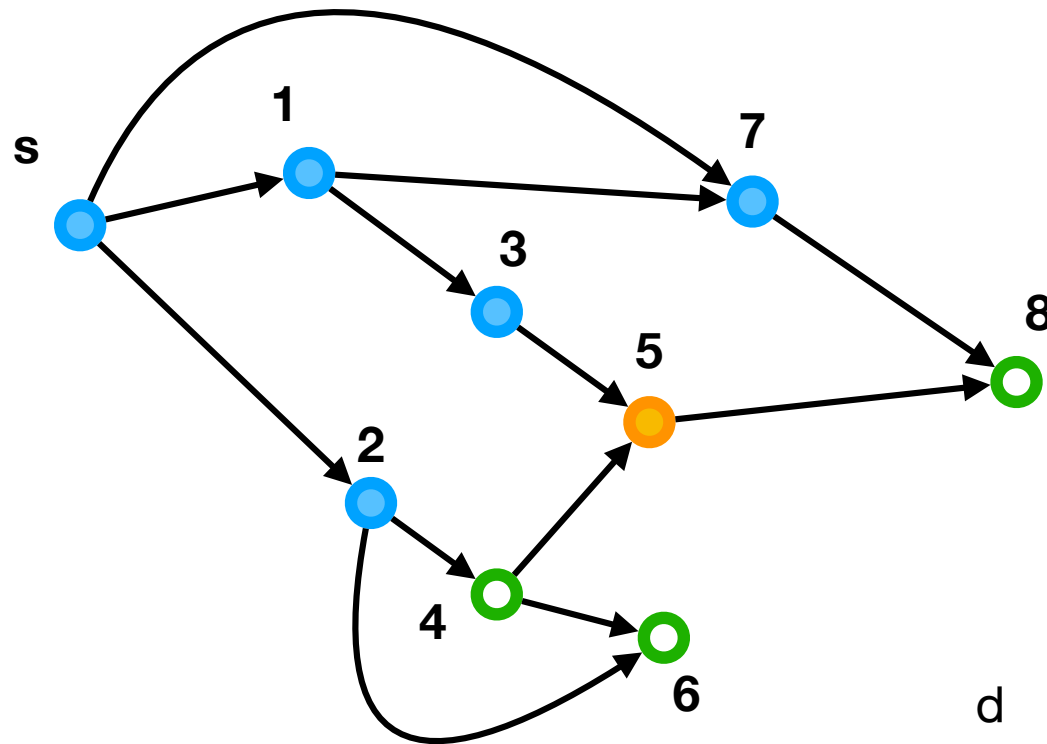






- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	$+\infty$	2	1	2
3	4	6	8					

queue Q

Shortest-Path-BFS: Example

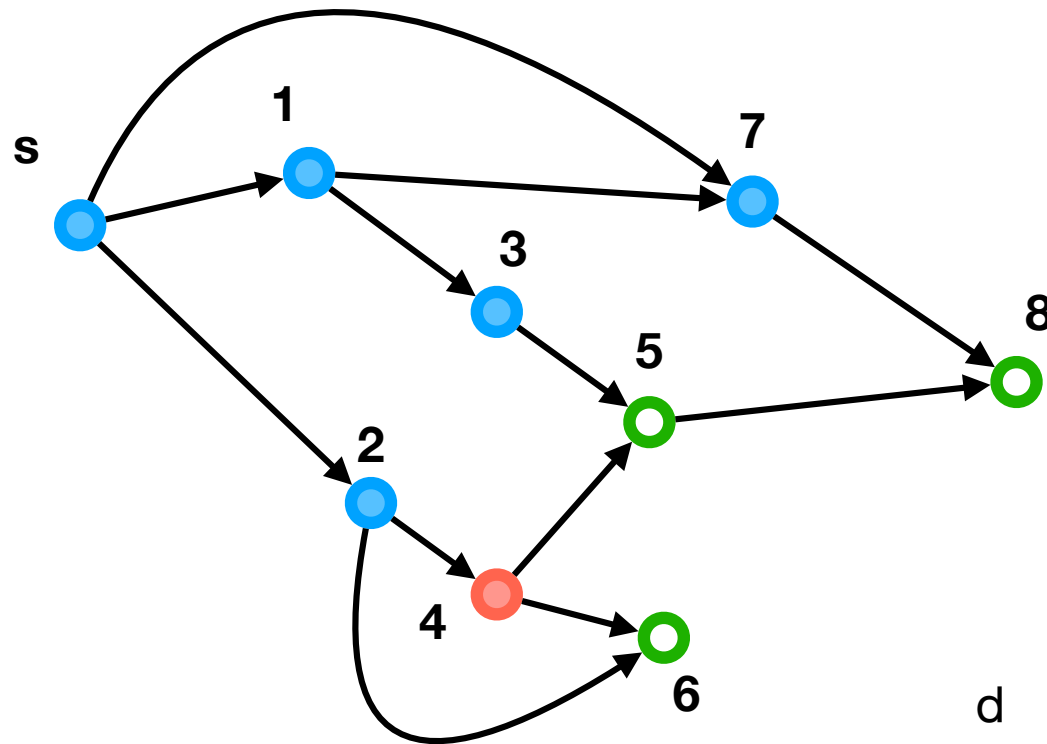


-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
4	6	8	5					

queue Q

Shortest-Path-BFS: Example

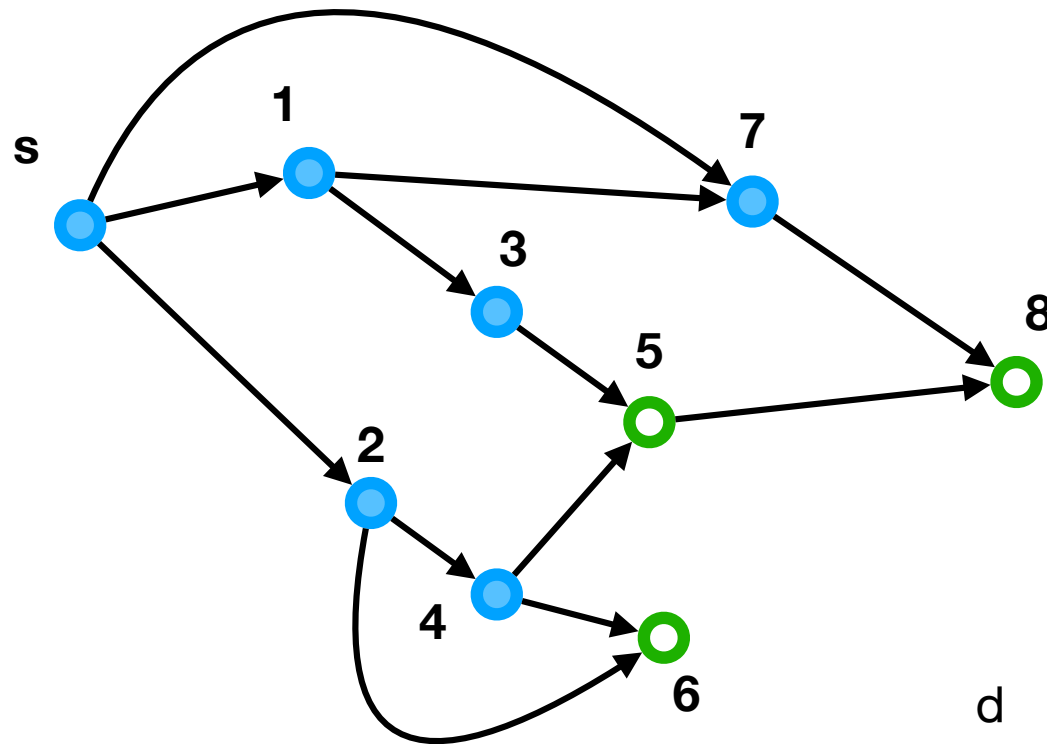






- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
4	6	8	5					

queue Q

Shortest-Path-BFS: Example

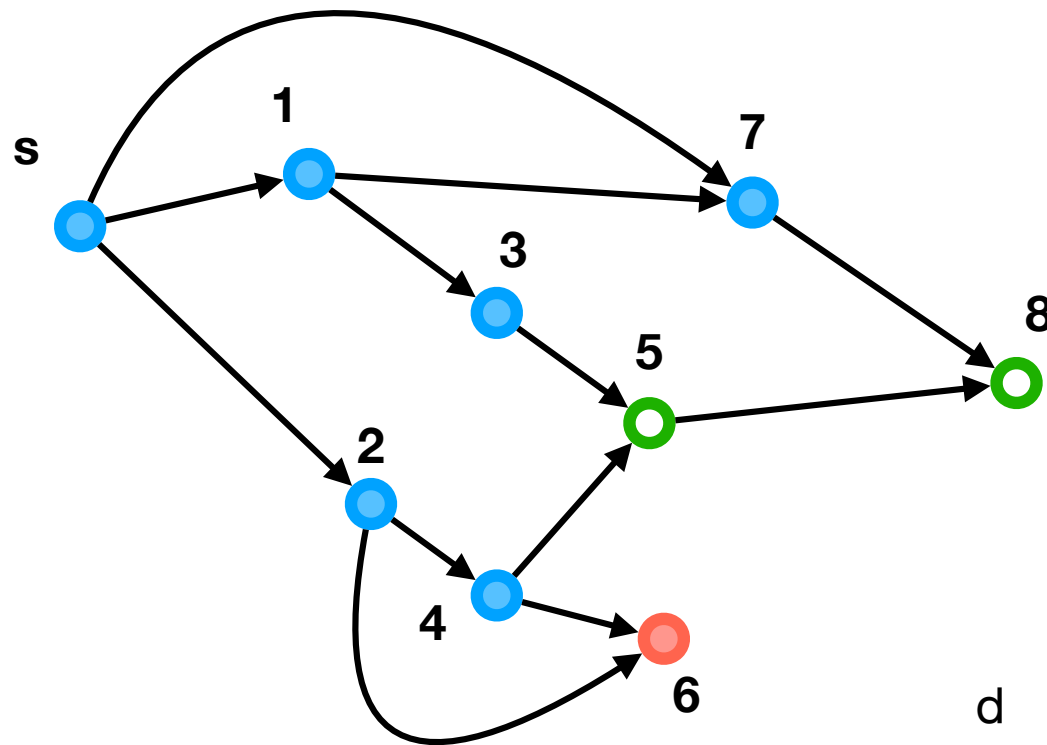


-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
6	8	5						

queue Q

Shortest-Path-BFS: Example

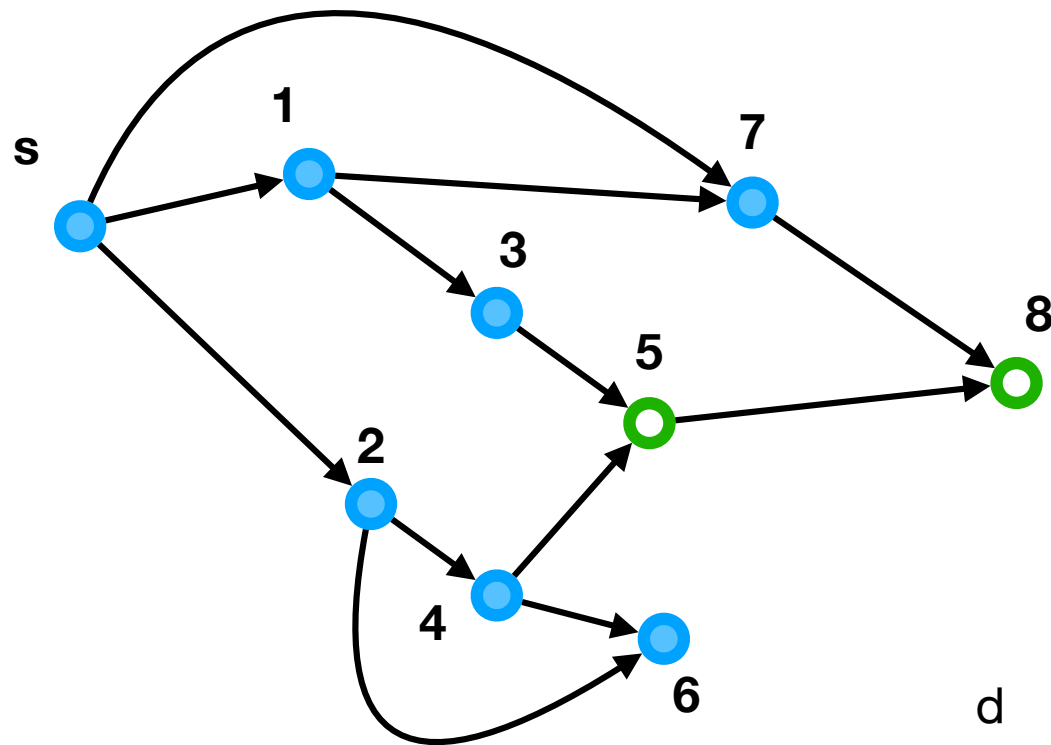






- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
6	8	5						

queue Q

Shortest-Path-BFS: Example

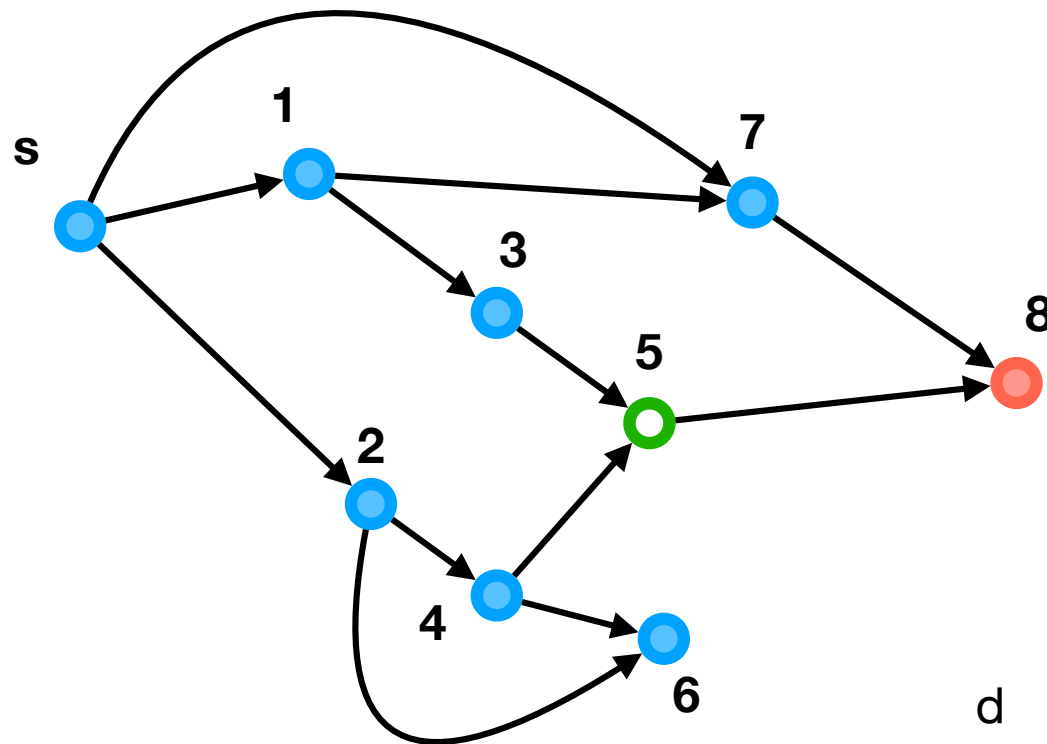






-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
8	5							

queue Q

Shortest-Path-BFS: Example

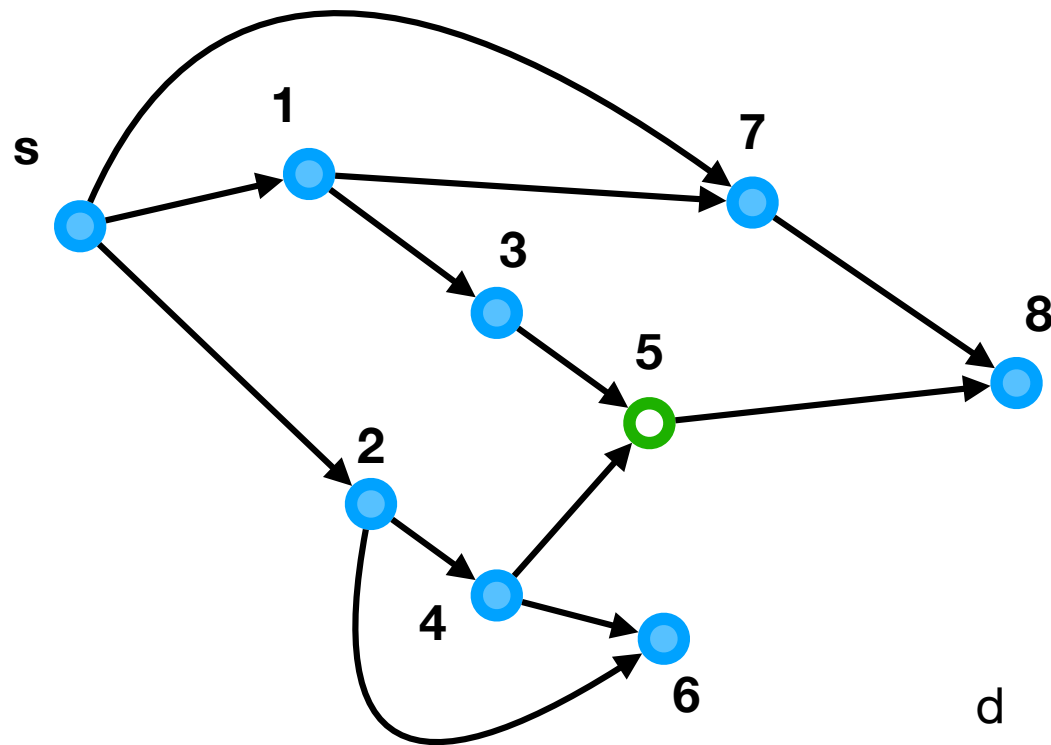






-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
8	5							

queue Q

Shortest-Path-BFS: Example

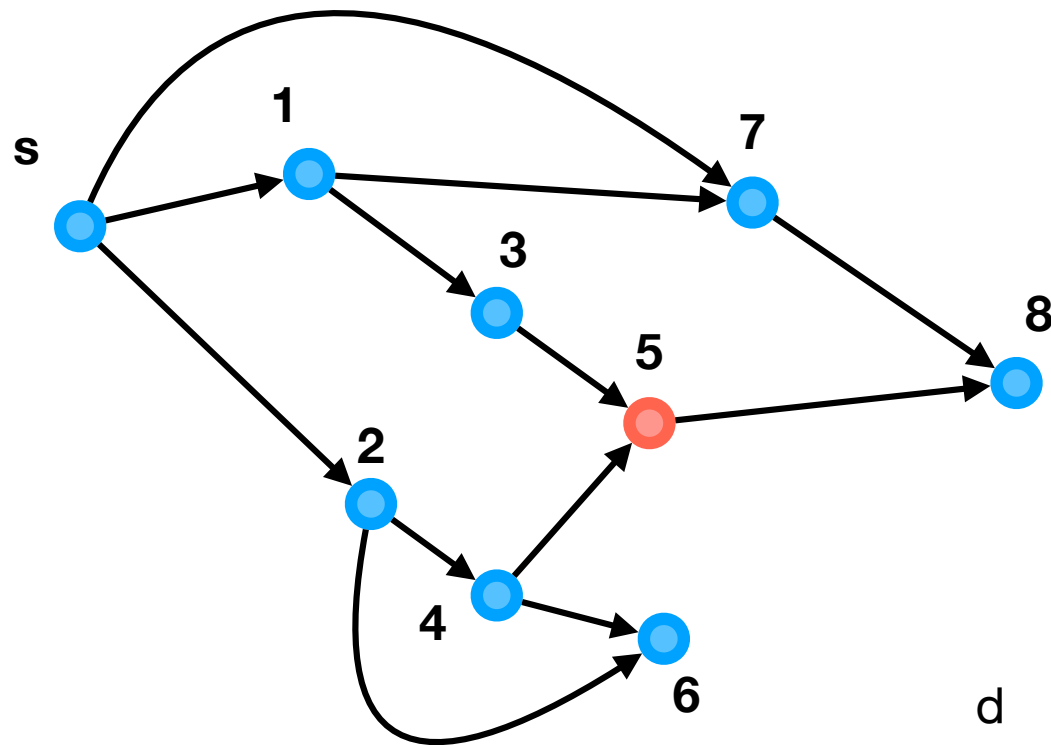


-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
5								

queue Q

Shortest-Path-BFS: Example

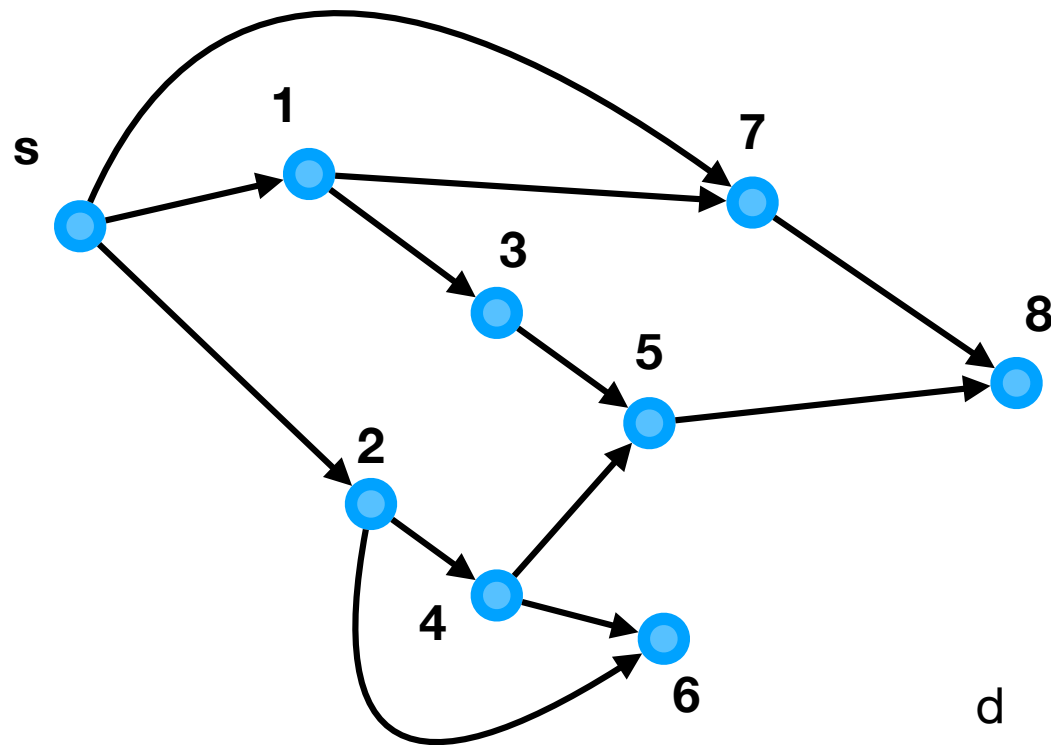






- “marked” vertices
- unmarked vertices
- current vertex in while-loop
- vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2
5								

queue Q

Shortest-Path-BFS: Example



-  “marked” vertices
-  unmarked vertices
-  current vertex in while-loop
-  vertices to be added to Q

s	1	2	3	4	5	6	7	8
0	1	1	2	2	3	2	1	2

queue Q

Shortest-Path-BFS

- Initialize $d[1:n] = +\infty$ for every vertex v
- Create a queue data structure Q and insert s to Q ; set $d[s] = 0$
- While Q is not empty;
 - Remove u from the top of the queue
 - For any $v \in N(u)$:
 - If $d[v] \neq +\infty$ continue to the next neighbor
 - Otherwise, let $d[v] = d[u] + 1$ and add v to Q

Proof of Correctness

Runtime Analysis

- Initialize $d[1:n] = +\infty$ for every vertex v
- Create a queue data structure Q and insert s to Q ; set $d[s] = 0$
- While Q is not empty;
 - Remove u from the top of the queue
 - For any $v \in N(u)$:
 - If $d[v] \neq +\infty$ continue to the next neighbor
 - Otherwise, let $d[v] = d[u] + 1$ and add v to Q
- Asymptotically the same as the original BFS
- So $O(n+m)$