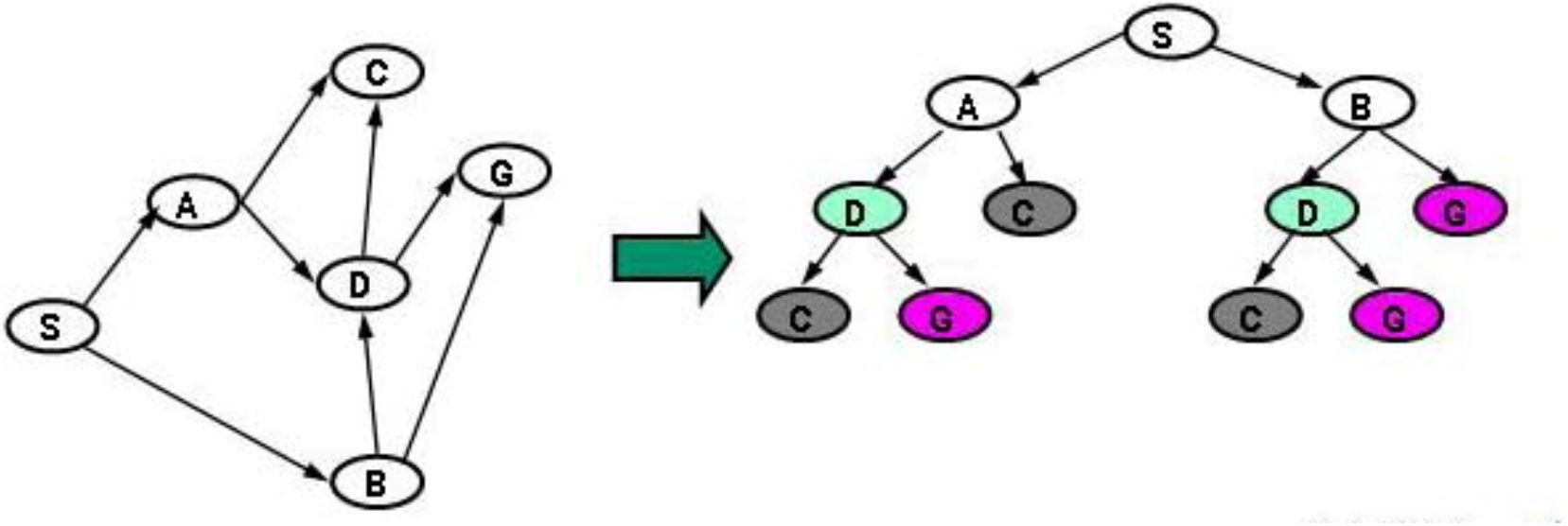
Review of BFS and DFS

305450 Lab01

Graph search as Tree search

- Trees are directed graphs without cycles
- We can turn graph search problems into tree search problems by
  - Replacing undirected links by 2 directed links
  - Avoiding loops in paths (or keeping track of visited nodes)



Terminology

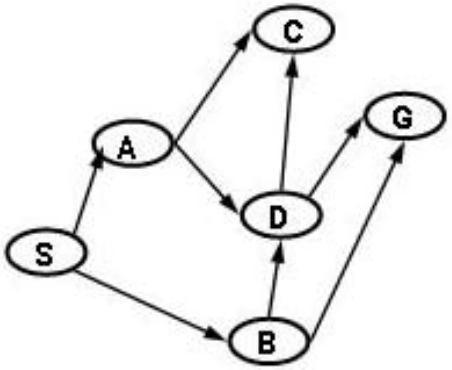
- Visited – a state N is first visited when a path to N first gets added to Q.
  - We visited a state when place it on Q, but we have not yet generated its descendants
- Expanded – a state N is expanded when it is the state of a node that is pulled of Q.
  - At this point, the descendants of N are visited and the path that led to M is extended to the descendants.
  - When a node M is expanded, we will not expand it again, hence we discard it from Q.

Depth First Search

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1		
2		
3		
4		
5		

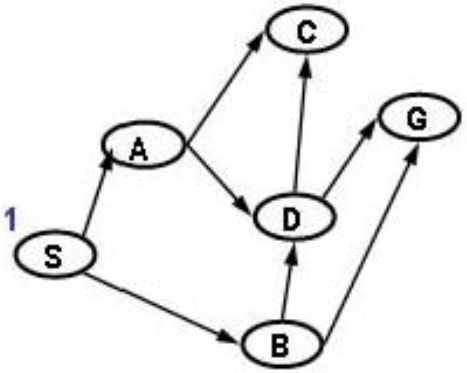


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1	(S)	S
2		
3		
4		
5		

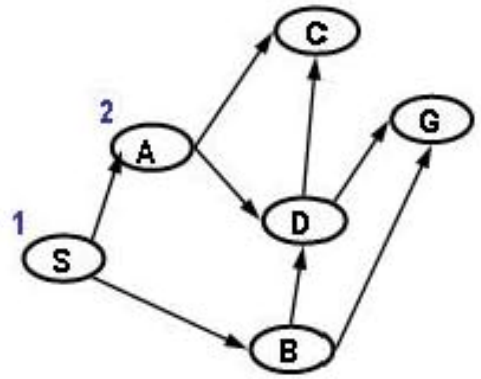


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A, B, S
3		
4		
5		

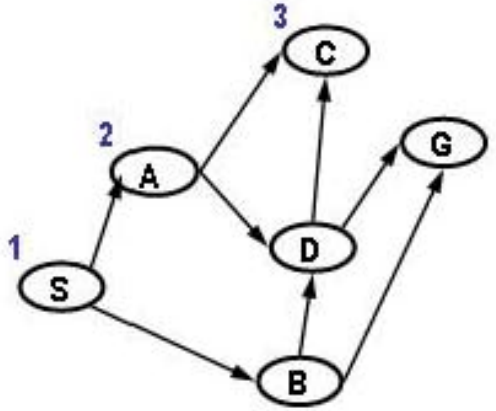


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A, B, S
3	(C A S) (D A S) (B S)	C,D,B,A,S
4		
5		

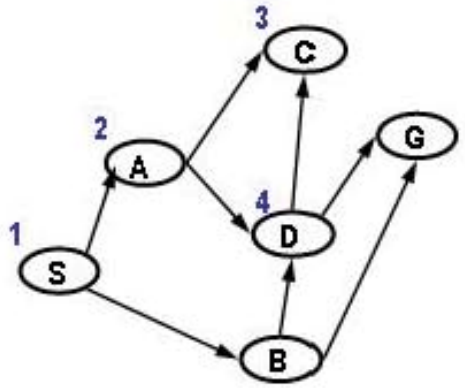


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A, B, S
3	(C A S) (D A S) (B S)	C,D,B,A,S
4	(D A S) (B S)	C,D,B,A,S
5		

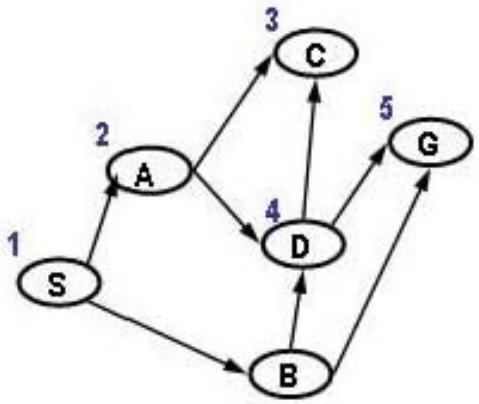


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A, B, S
3	(C A S) (D A S) (B S)	C,D,B,A,S
4	(D A S) (B S)	C,D,B,A,S
5	(G D A S) (B S)	G,C,D,B,A,S

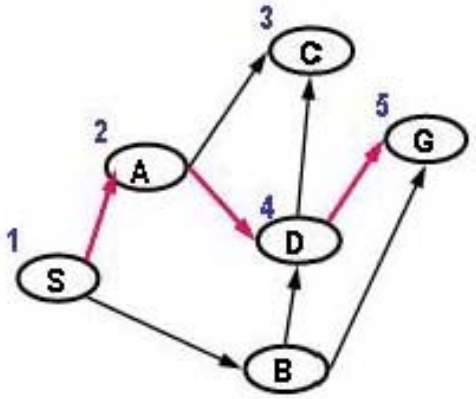


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

- Pick first element of Q; Add path extension to *front* of Q.

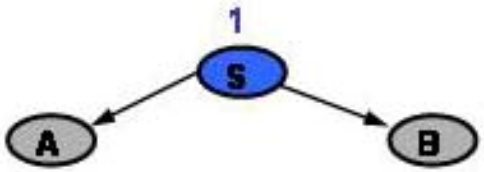
	Q	Visited
1	(S)	S
2	(A S) (B S)	A, B, S
3	(C A S) (D A S) (B S)	C,D,B,A,S
4	(D A S) (B S)	C,D,B,A,S
5	(G D A S) (B S)	G,C,D,B,A,S

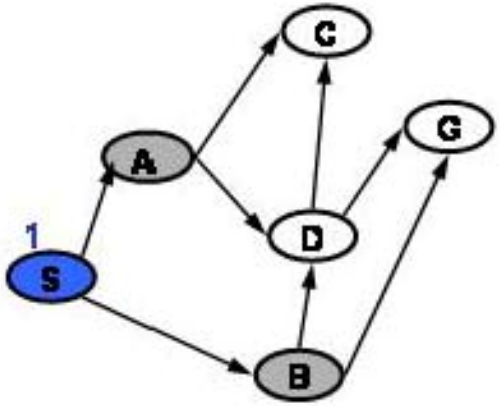


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Depth First Search

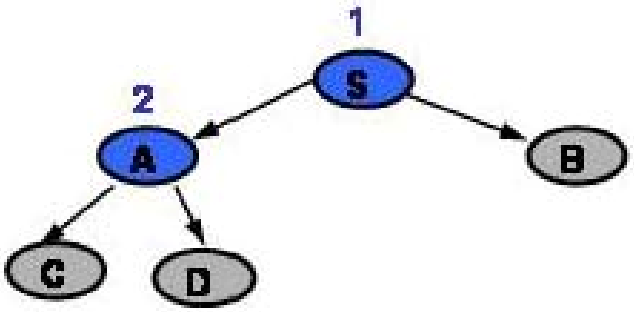
- Another way to see it (as tree)

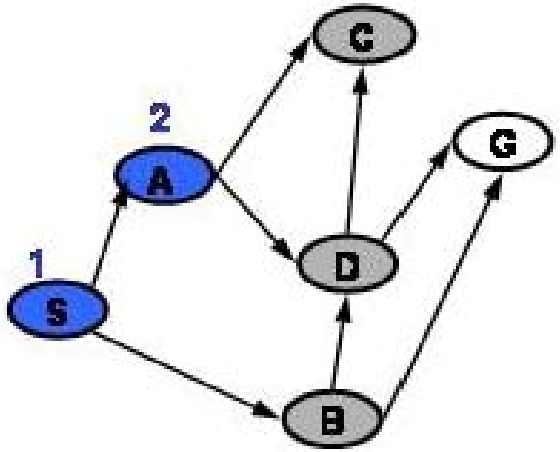




Depth First Search

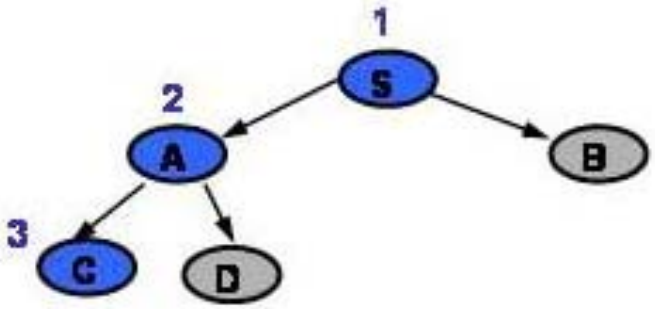
- Another way to see it (as tree)

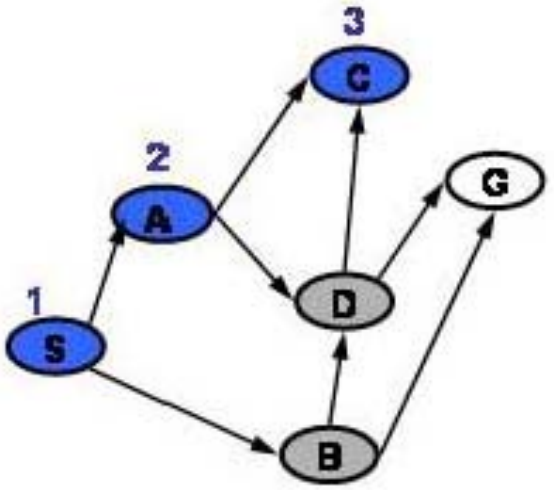




Depth First Search

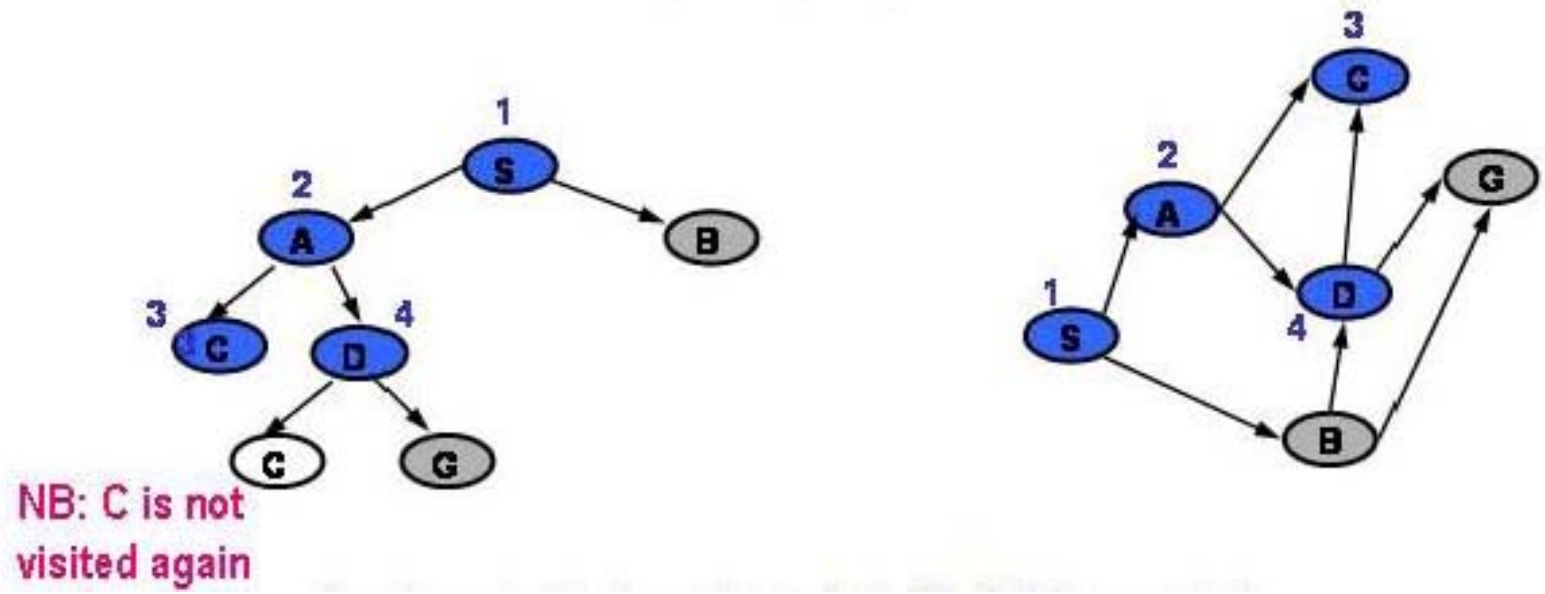
- Another way to see it (as tree)





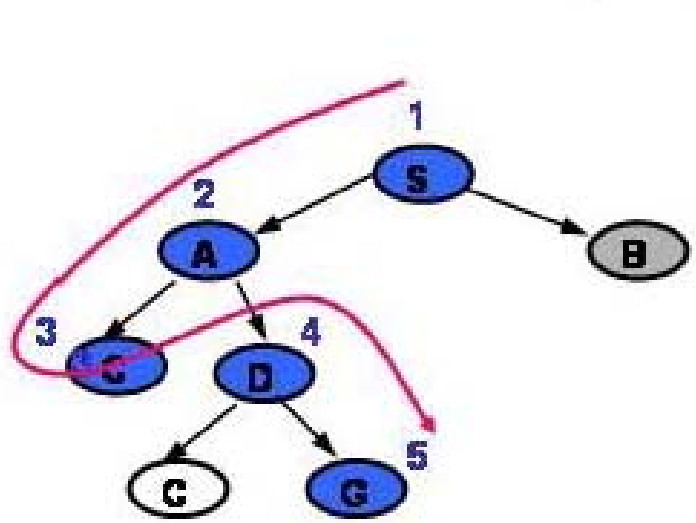
Depth First Search

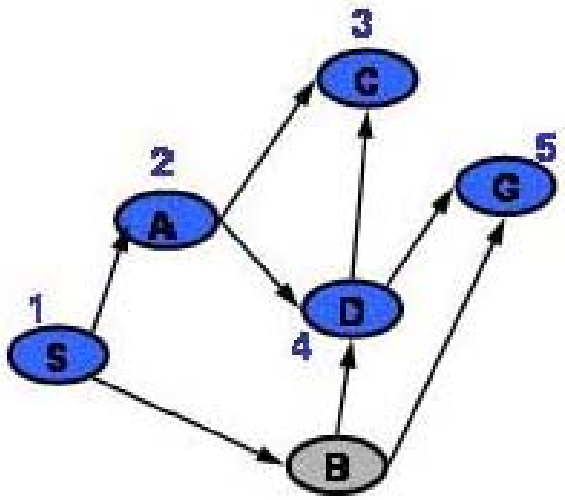
- Another way to see it (as tree)



Depth First Search

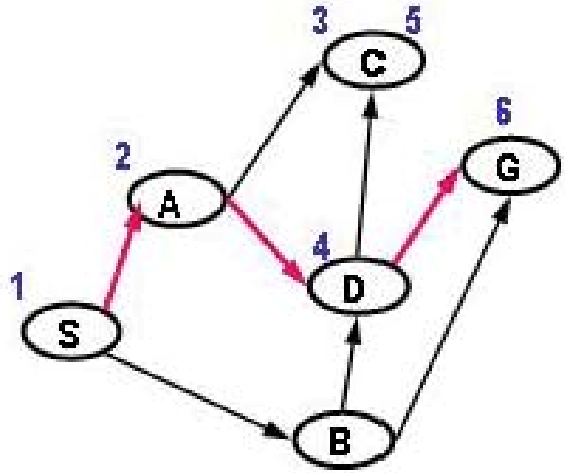
- Another way to see it (as tree)





Depth First Search (without Visited list)

	Q
1	(S)
2	(A S) (B S)
3	(C A S) (D A S) (B S)
4	(D A S) (B S)
5	(C D A S) (G D A S) (B S)
6	(G D A S) (B S)

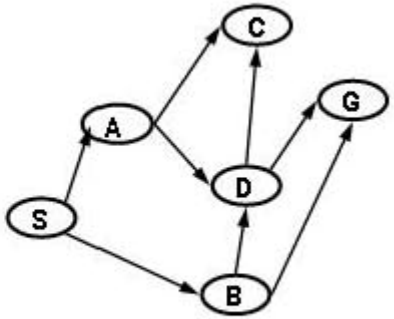


Breath First Search

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2		
3		
4		
5		
6		

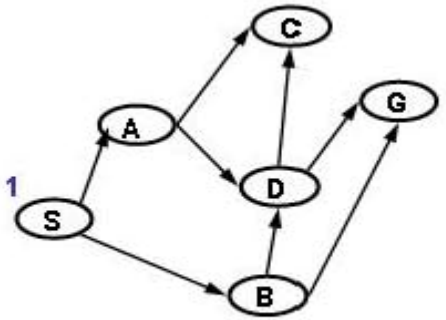


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3		
4		
5		
6		

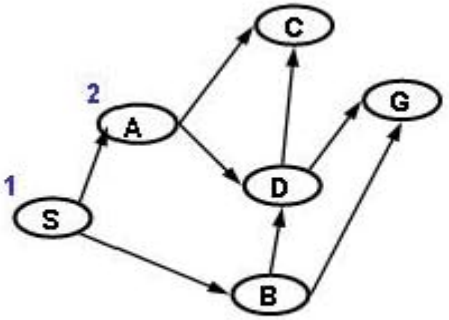


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4		
5		
6		

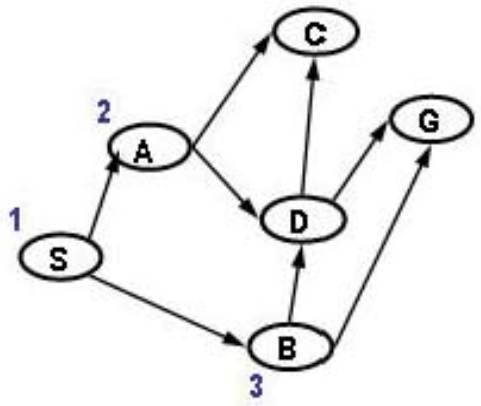


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5		
6		

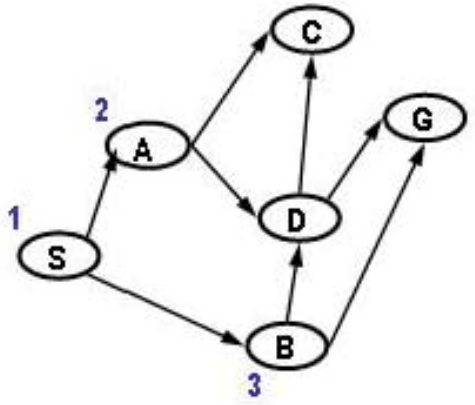


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5		
6		

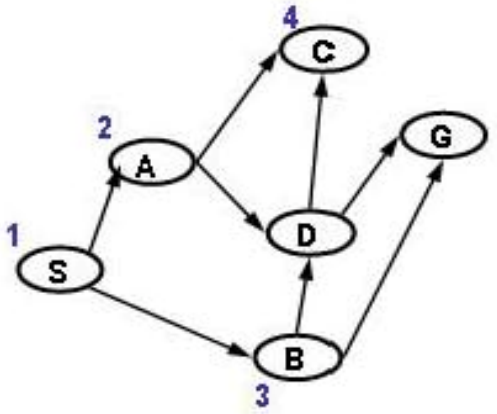


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5	(D A S) (G B S)	G,C,D,B,A,S
6		

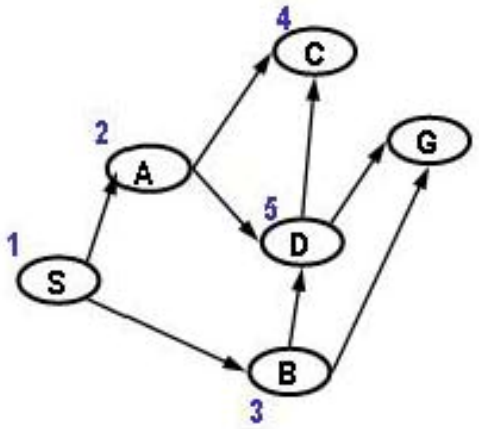


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5	(D A S) (G B S)	G,C,D,B,A,S
6		

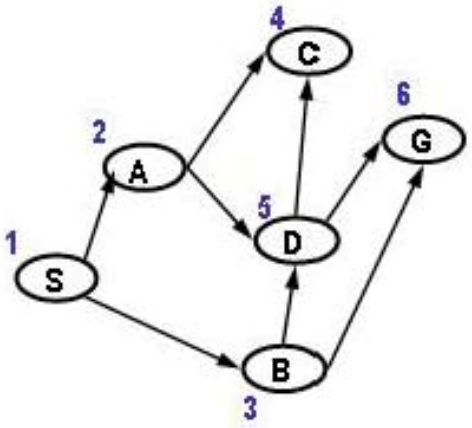


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5	(D A S) (G B S)	G,C,D,B,A,S
6	(G B S)	G,C,D,B,A,S

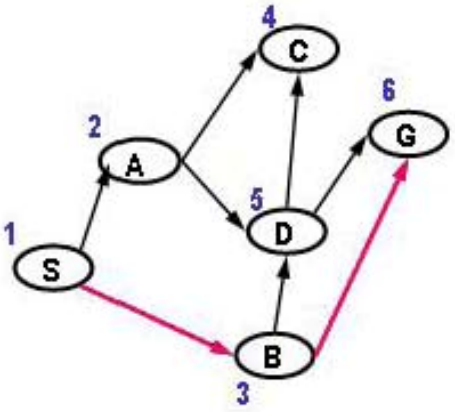


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

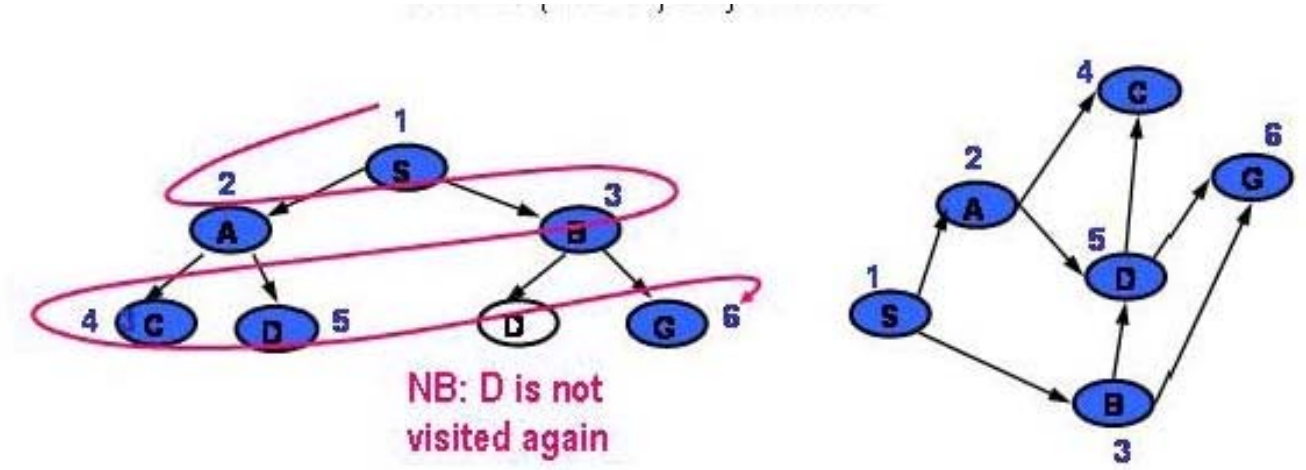
	Q	Visited
1	(S)	S
2	(A S) (B S)	A,B,S
3	(B S) (C A S) (D A S)	C,D,B,A,S
4	(C A S) (D A S) (G B S)*	G,C,D,B,A,S
5	(D A S) (G B S)	G,C,D,B,A,S
6	(G B S)	G,C,D,B,A,S



- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search

- Pick first element of Q; Add path extension to end of Q.

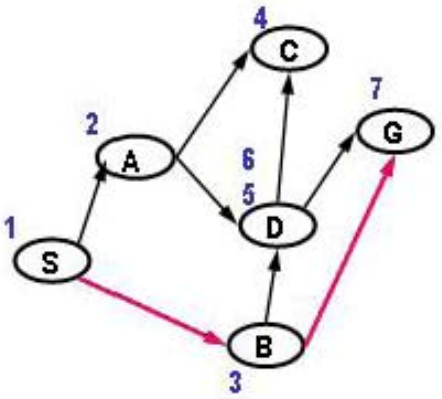


- Added path is in blue
- We show the paths in reversed order; the node's state is the first element

Breath First Search (without Visited list)

- Pick first element of Q; Add path extension to end of Q.

	Q
1	(S)
2	(A S) (B S)
3	(B S) (C A S) (D A S)
4	(C A S) (D A S) (D B S) (G B S)*
5	(D A S) (D B S) (G B S)
6	(D B S) (G B S) (C D A S) (G D A S)
7	(G B S) (C D A S) (G D A S) (C D B S) (G D B S)



- Added path is in blue
- We show the paths in reversed order; the node's state is the first element
- We could have stopped here, when the fist path to the goal was generated