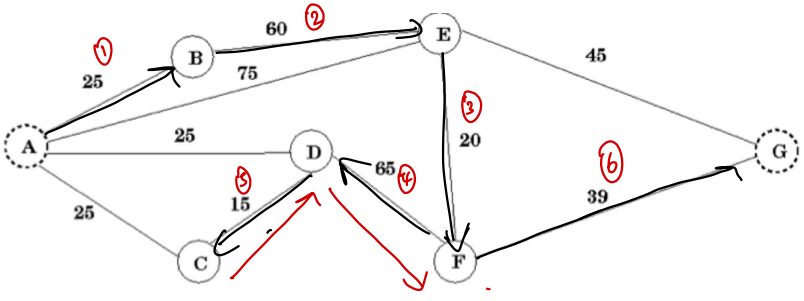
Depth-First Search.



legend

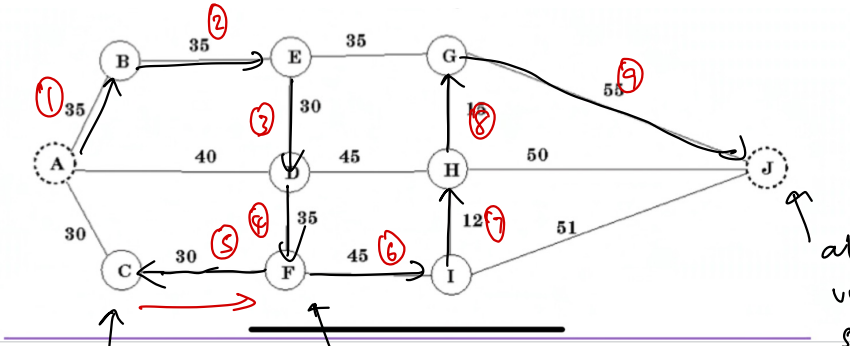
← Backtrack

Queue:

- Add A
- Add B
- Add E
- Add F
- Add D
- Add C
- Add G.

Stack:

- push A
- push B
- push F
- push D
- push C
- pop C ← start backtracking, no new node
- pop D
- pop f
- push G ← theres still an unvisited node on f node.
- pop G
- pop E
- pop B
- pop A.

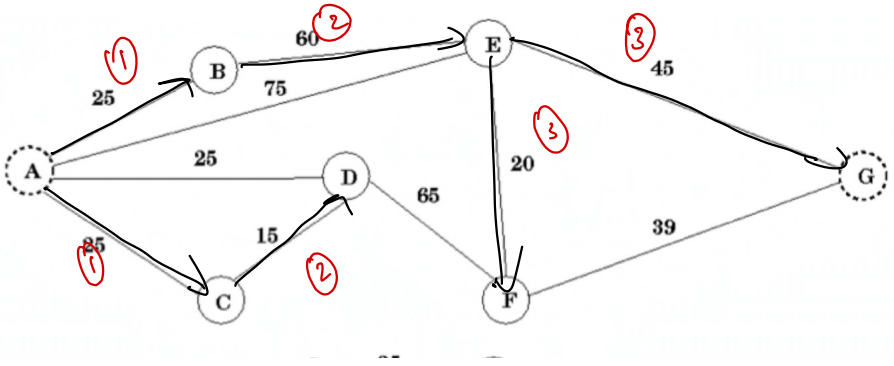


queue:

- Add A
- Add B
- Add E
- Add D
- Add F
- Add C
- Add I
- Add H
- Add G
- Add J.

Stack

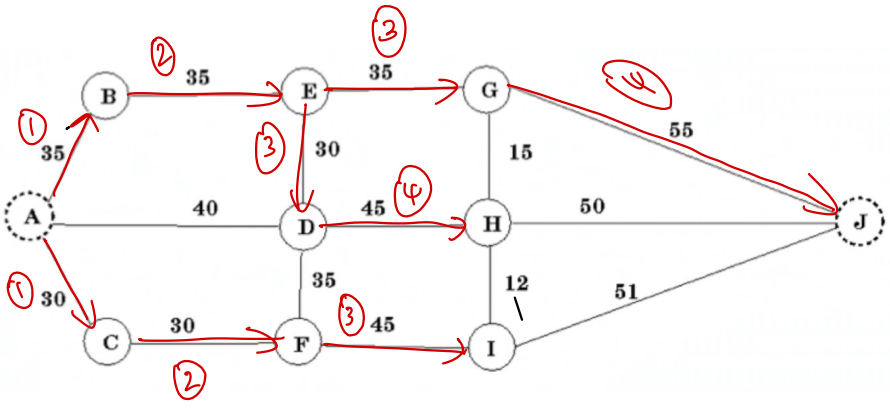
- Push A
- Push B
- Push E
- Push D
- Push F
- Push C
- Pop C
- Pop F ← still has one unvisited node
- push I
- push H
- push G
- push J
- pop J
- pop G
- pop H
- pop I
- pop F
- pop D
- pop E
- pop B
- pop A.



BFS Tree	queue
- Add vertex A	enqueue A dequeue A
- Add vertex B - Add edge (A,B) - Add vertex C - Add edge (A,C)	enqueue B enqueue C
- Add vertex E - Add edge (B,E)	dequeue B enqueue E
- Add vertex D - Add edge (C,D)	dequeue C enqueue D
- Add vertex F - Add edge (E,F) - Add vertex G - Add edge (E,G)	dequeue E enqueue F enqueue G

dequeue D
dequeue F
dequeue G

no more unvisited nodes.



BFS Tree	queue
Add vertex A	enqueue A
Add vertex B Add edge(A,B)	dequeue A. enqueue B
Add vertex C Add edge(A,C)	enqueue C
Add vertex E Add edge(B,E)	dequeue B enqueue E
Add vertex F Add edge(C,F)	dequeue C enqueue F.

Add vertex D
Add edge(E,D)
Add vertex G
Add edge(E,G)
	dequeue E
	enqueue D
	enqueue G

Add vertex I
Add edge (F, I)

dequeue F
enqueue I

Add vertex H
Add edge (D, H)

dequeue D
enqueue H.

Add vertex J
Add edge (G, J)

dequeue G
enqueue J

dequeue I
dequeue H
dequeue J

} no more unvisited nodes.