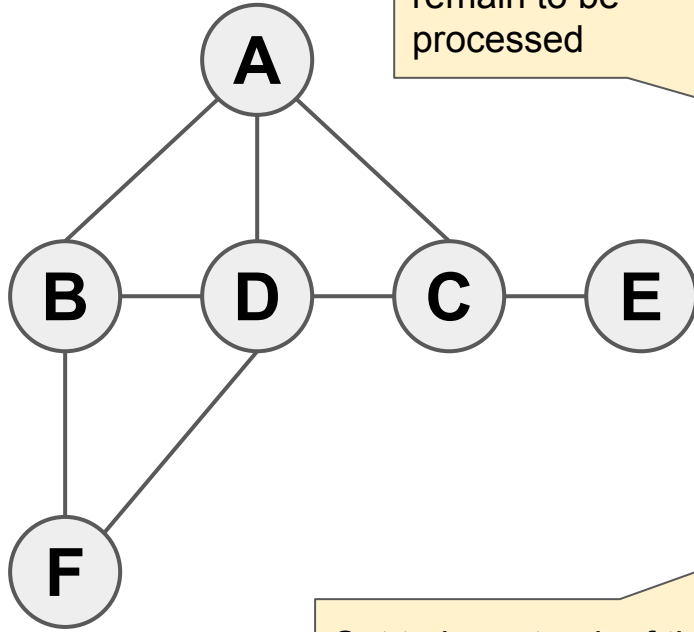
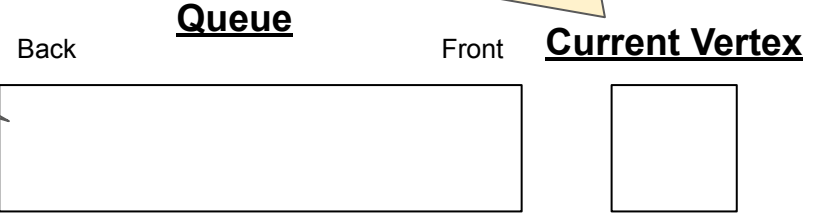


BFS Example

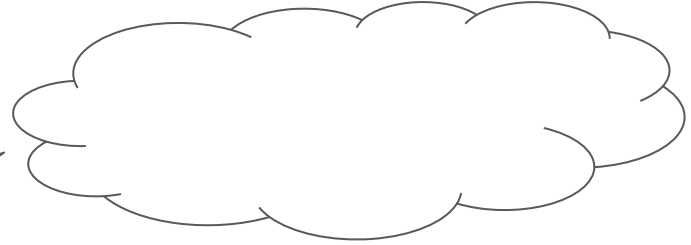


Queue to keep track of the vertices that remain to be processed

The current vertex we are processing



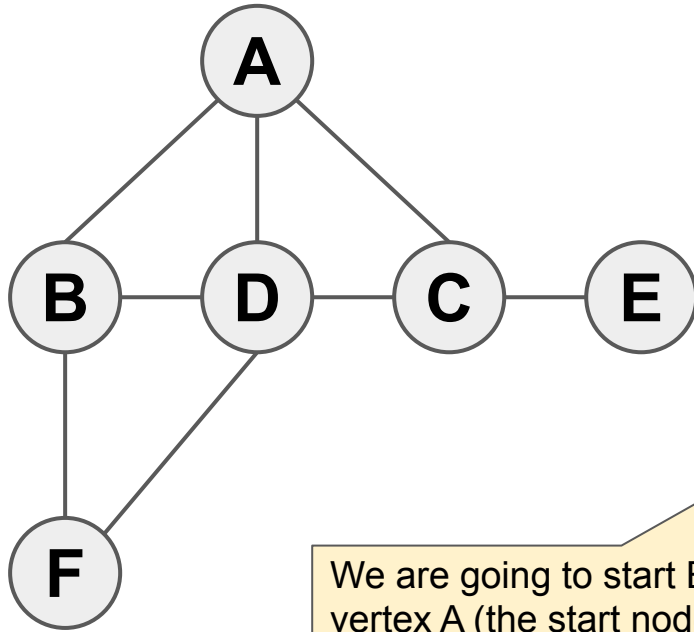
Explored



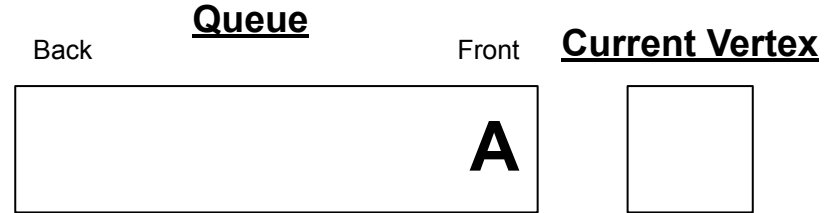
In this example, the processing of each vertex will just involve printing it to the screen.

Set to keep track of the list of vertices we have already encountered (or “explored”)

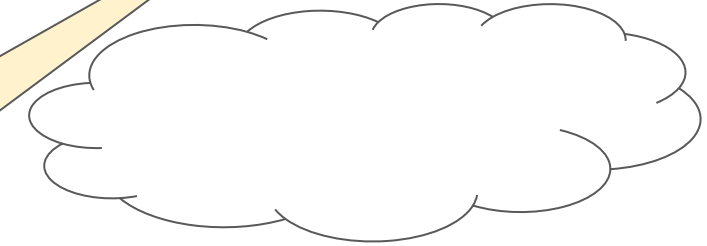
BFS Traversal:



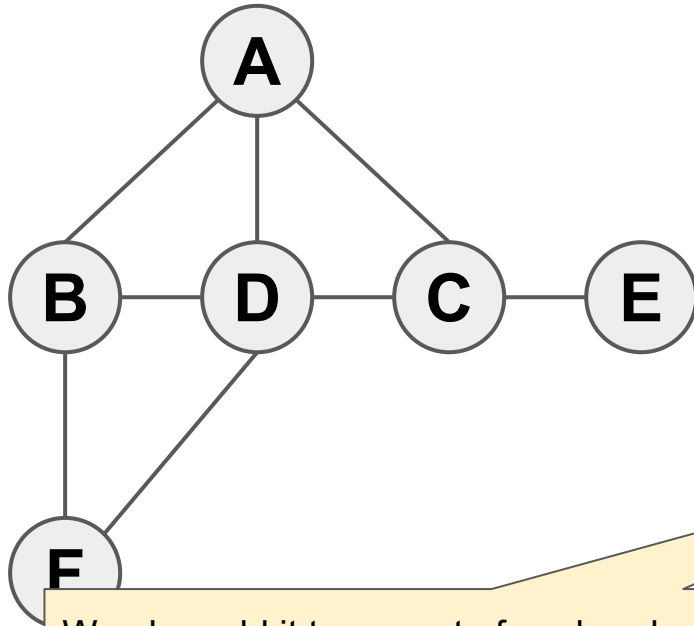
We are going to start BFS in vertex A (the start node is an input to the algorithm), so we enqueue that vertex in the queue



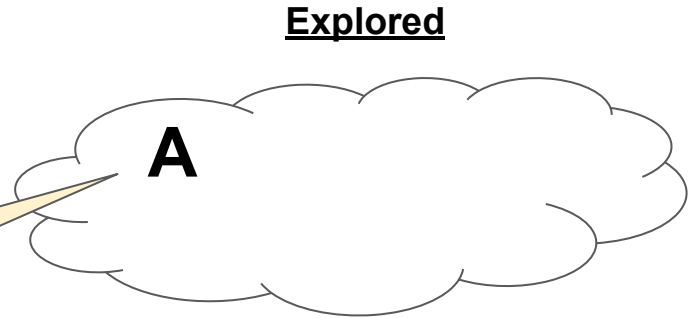
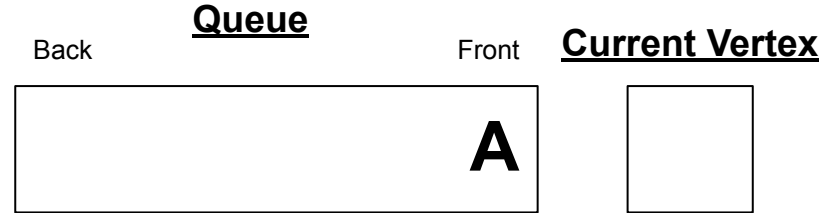
Explored



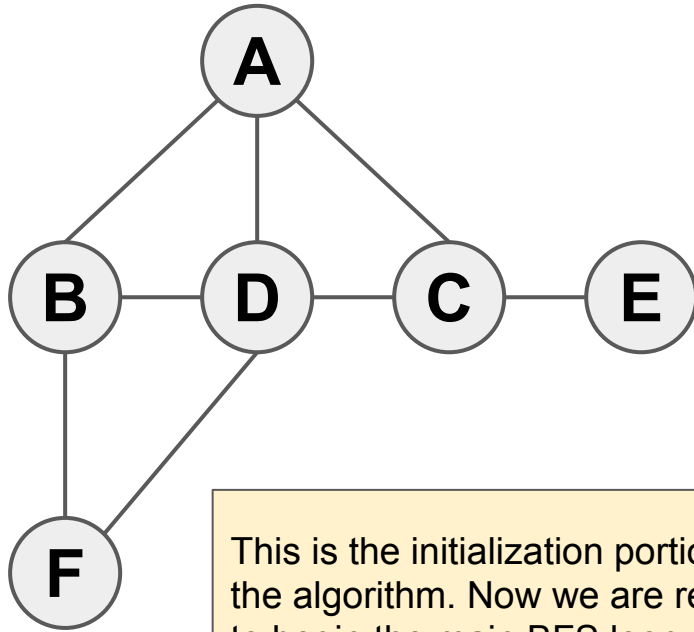
BFS Traversal:



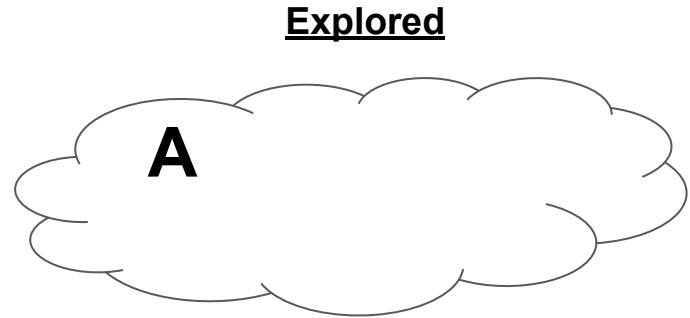
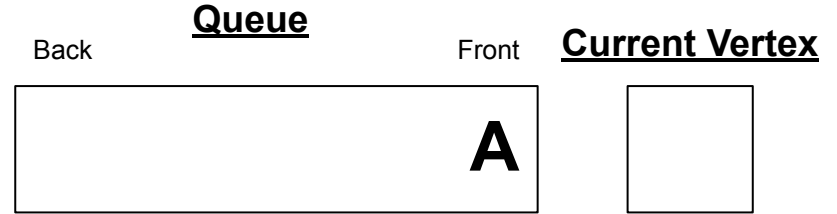
We also add it to our set of explored vertices. Notice how this keeps track of any vertex we have encountered, regardless of whether we have actually “visited” it (in the sense of processing that vertex in some way)



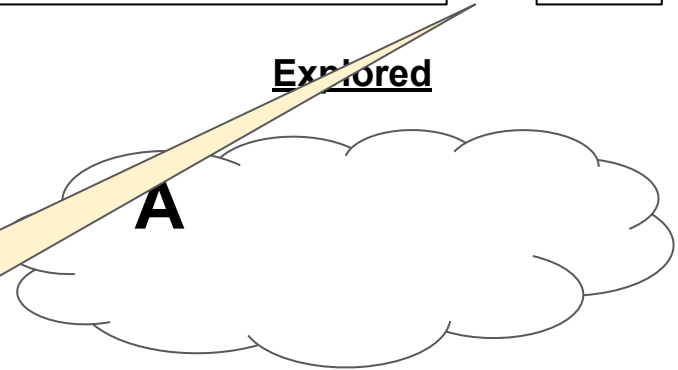
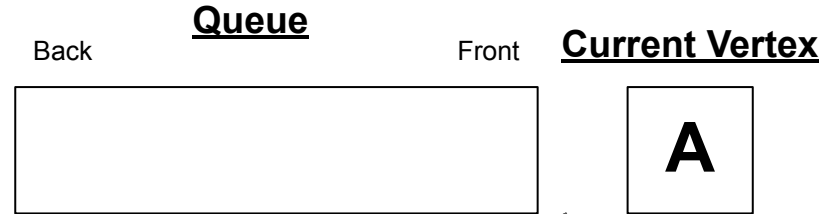
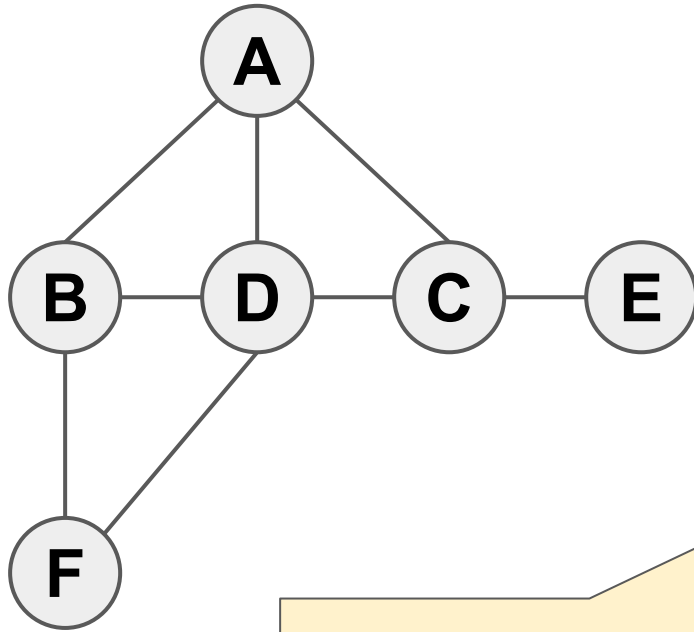
BFS Traversal:



This is the initialization portion of the algorithm. Now we are ready to begin the main BFS loop.

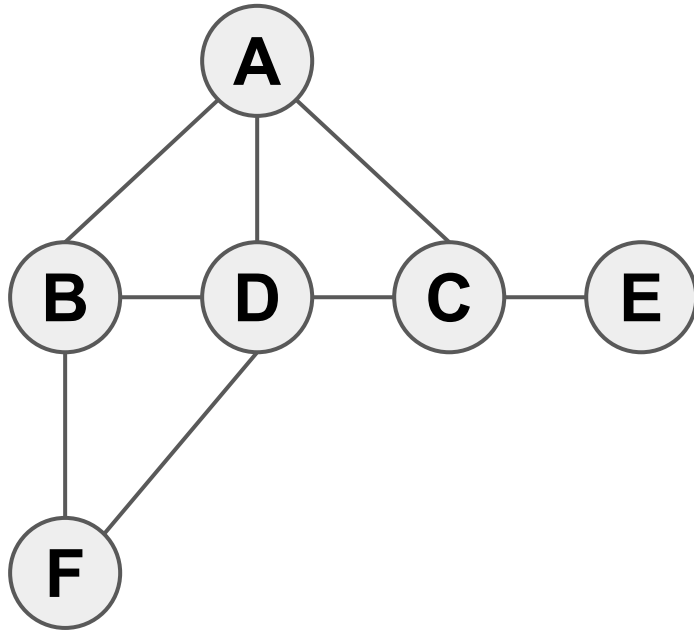


BFS Traversal:

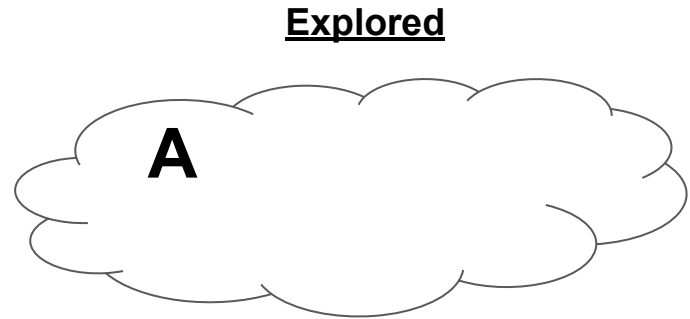
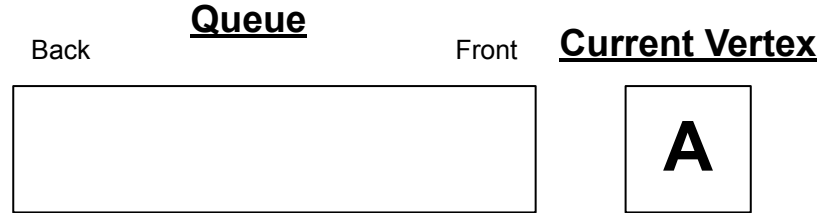


We dequeue A and it becomes our current vertex

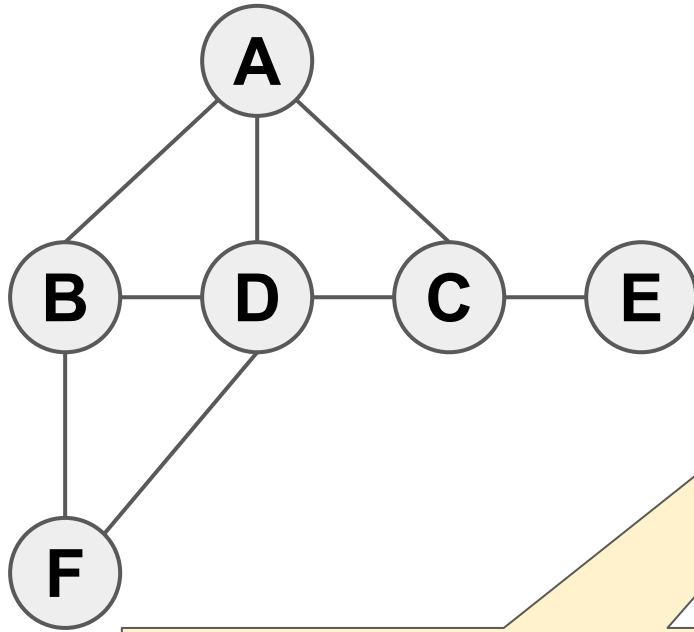
BFS Traversal:



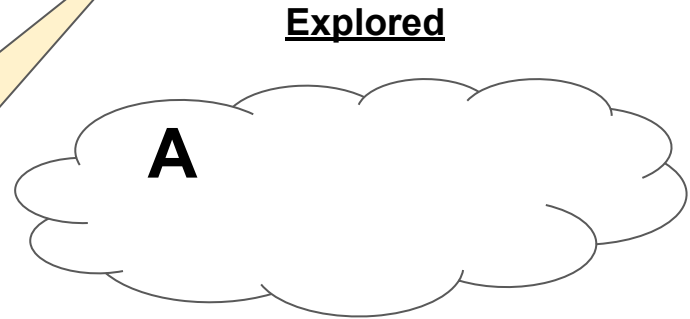
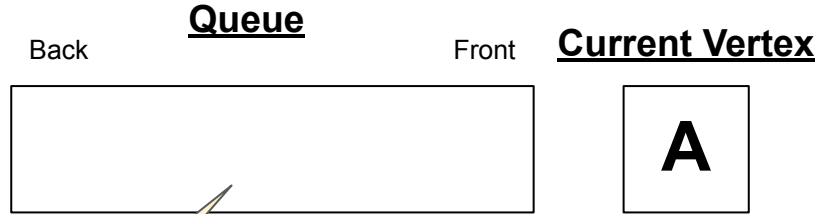
At this point, we process the current vertex in some way. We are just going to print it to the terminal.



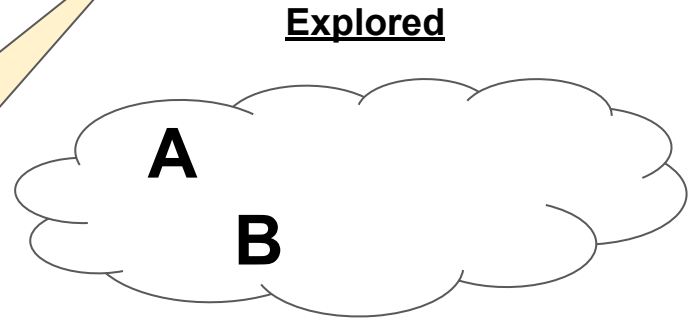
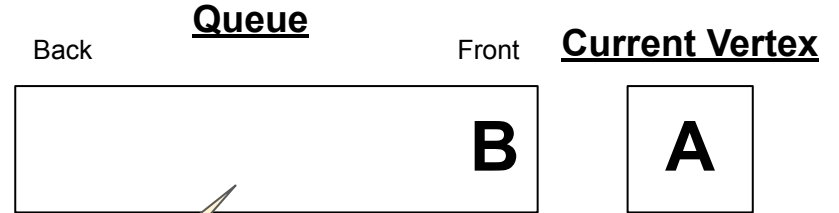
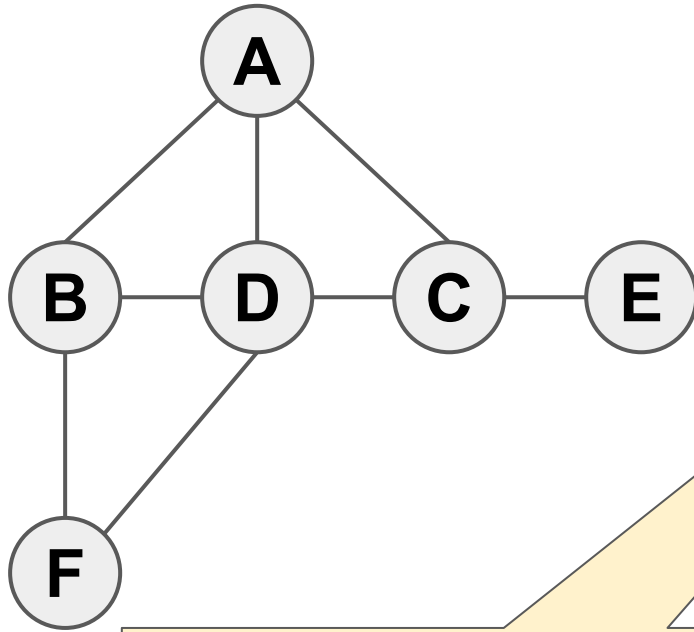
BFS Traversal: A



Next, we enqueue any *unexplored* neighbors of the current vertex, and mark them as explored (remember: this doesn't mean we've visited them yet; it means we've *encountered* them)

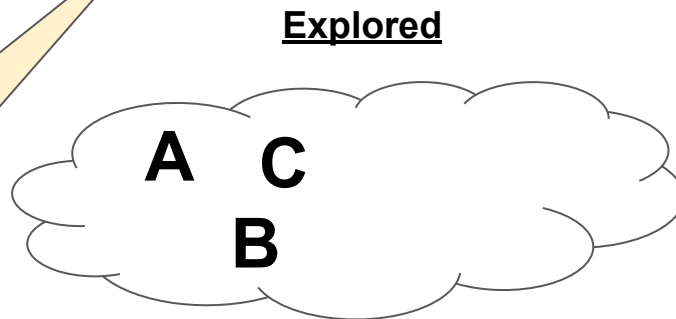
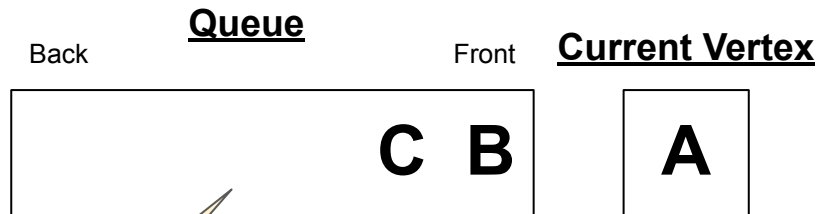
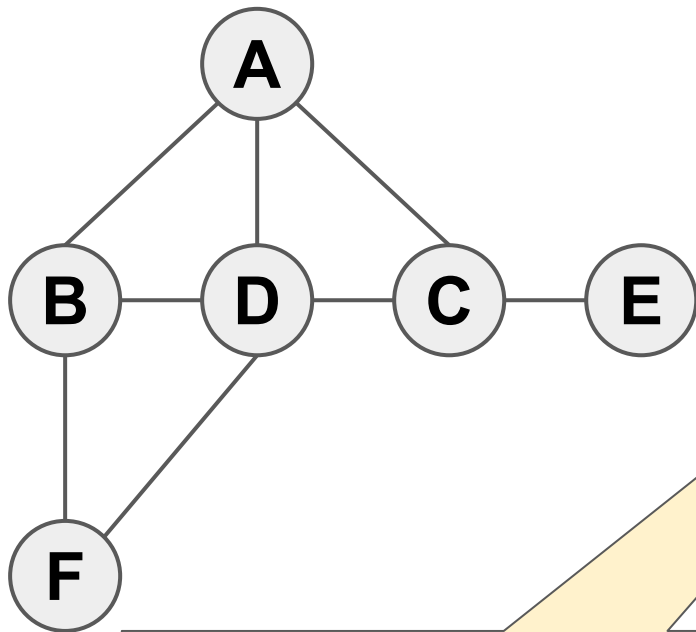


BFS Traversal: A



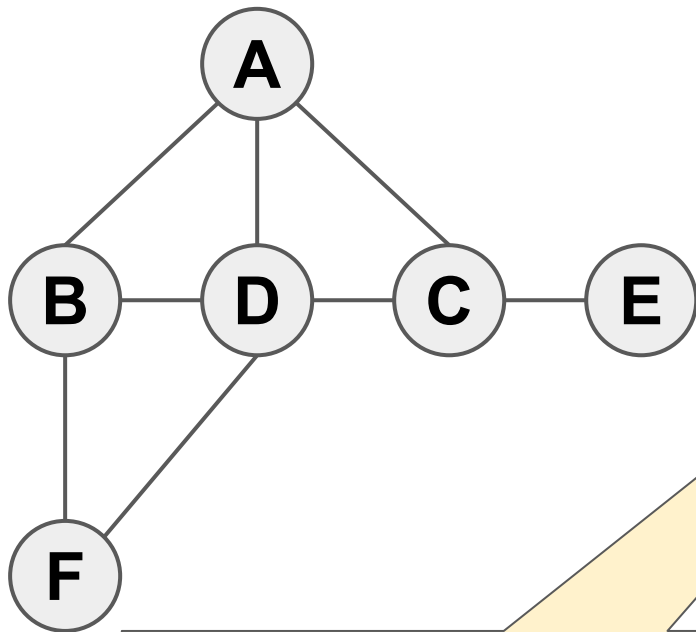
Next, we enqueue any *unexplored* neighbors of the current vertex, and mark them as explored (remember: this doesn't mean we've visited them yet; it means we've *encountered* them)

BFS Traversal: A

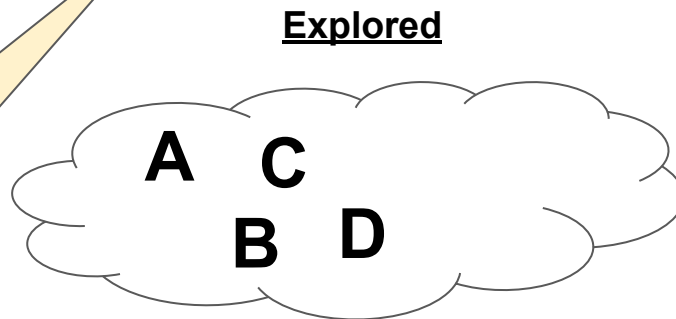
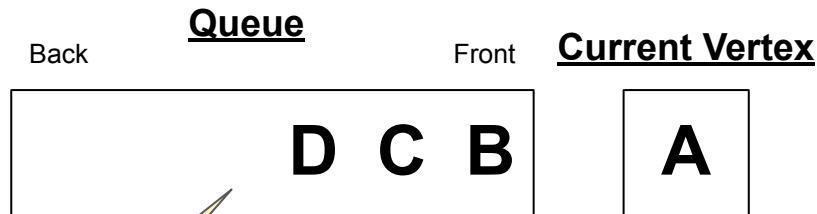


Next, we enqueue any *unexplored* neighbors of the current vertex, and mark them as explored (remember: this doesn't mean we've visited them yet; it means we've *encountered* them)

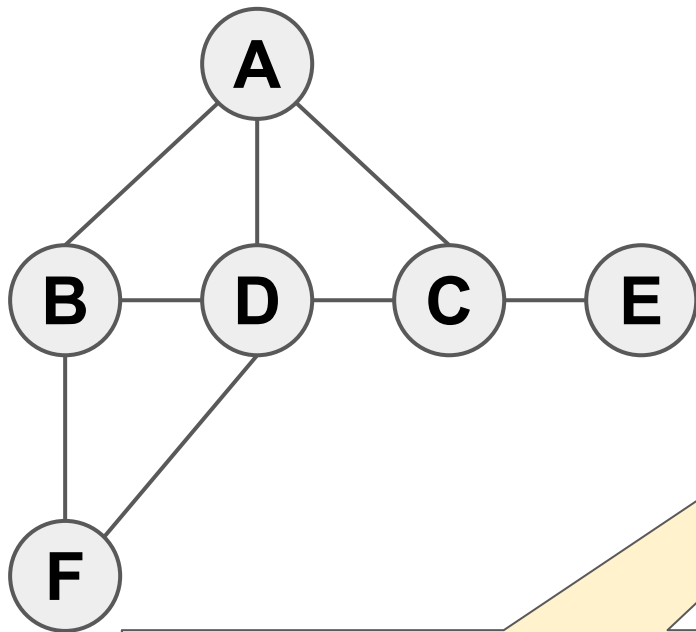
BFS Traversal: A



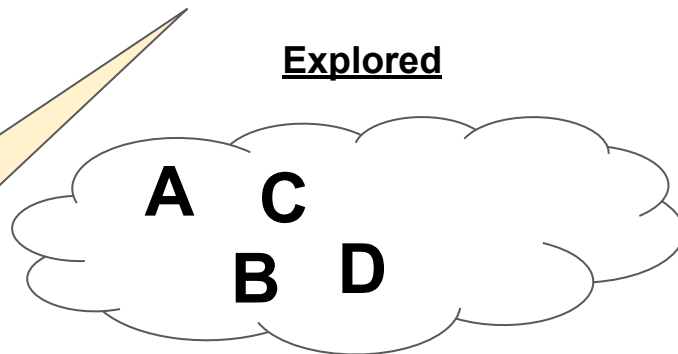
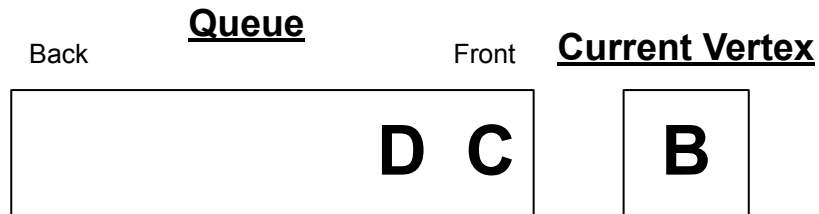
Next, we enqueue any *unexplored* neighbors of the current vertex, and mark them as explored (remember: this doesn't mean we've visited them yet; it means we've *encountered* them)



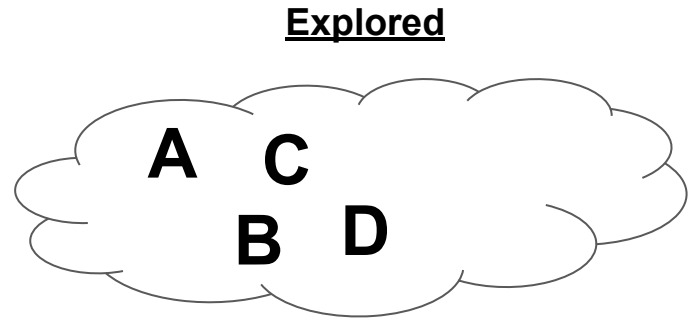
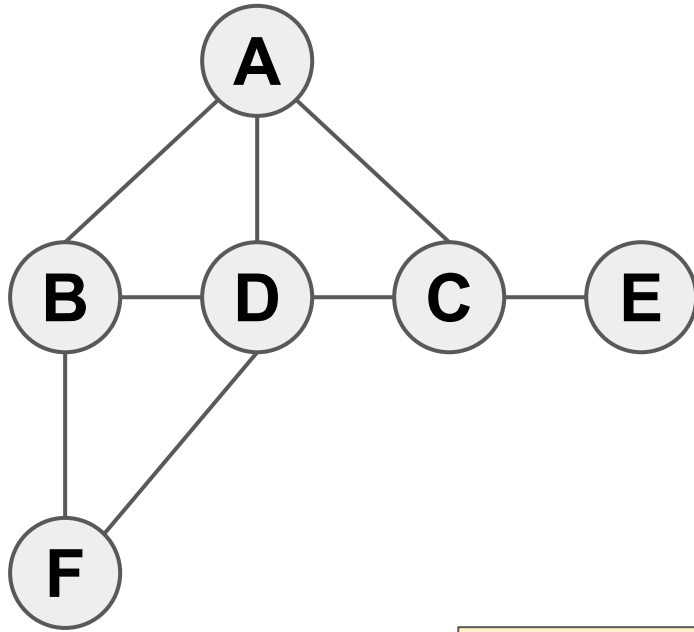
BFS Traversal: A



We dequeue B and it becomes our current vertex.

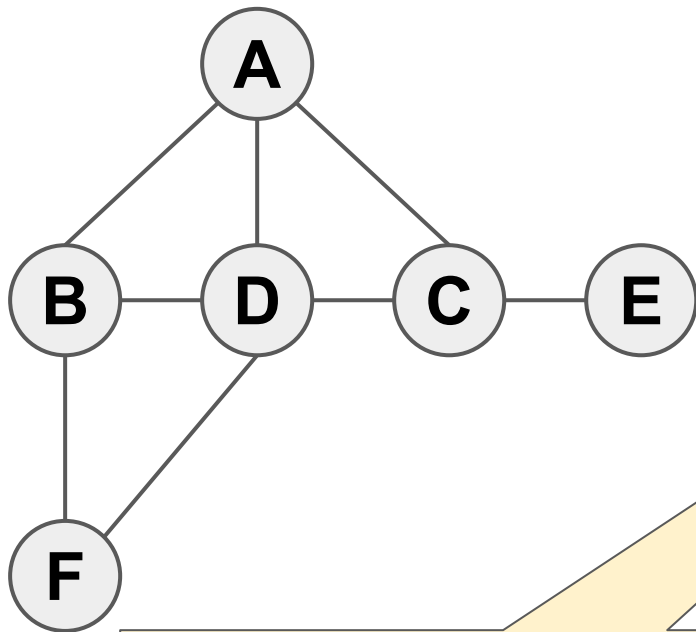


BFS Traversal: A

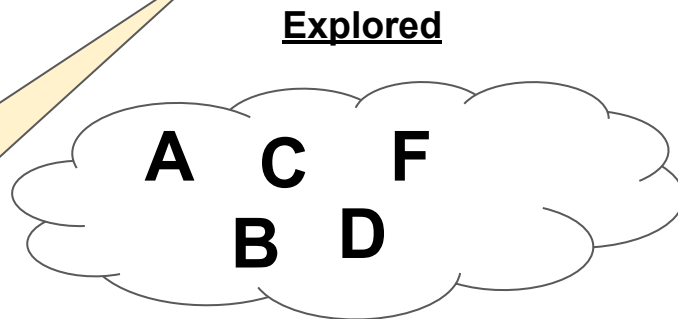
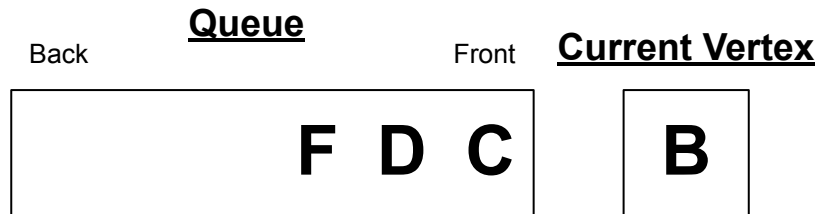


We process B.

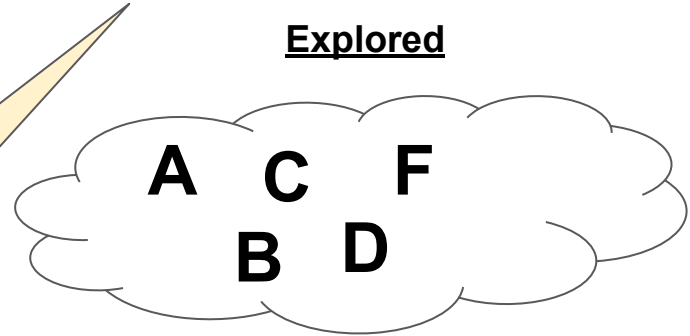
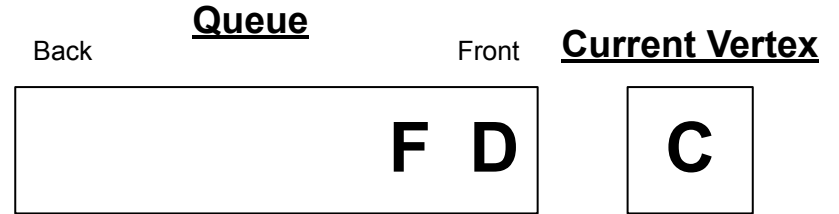
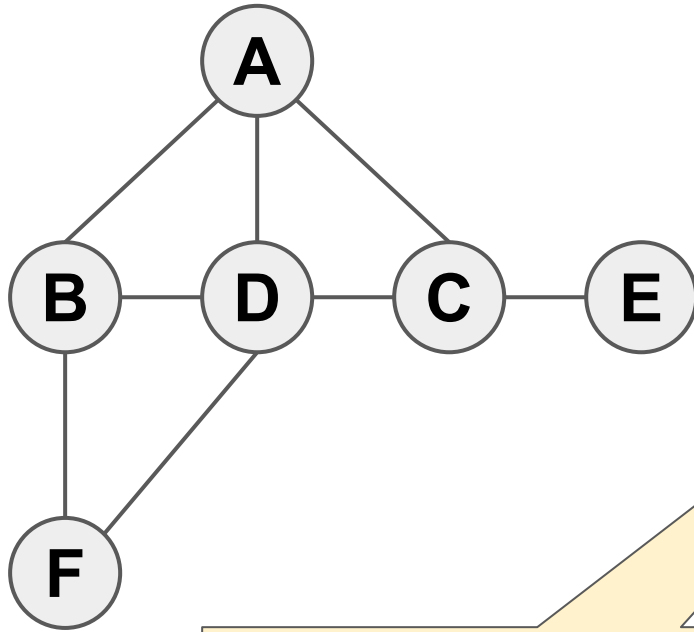
BFS Traversal: A B



And we enqueue all its unexplored neighbors. Notice how we *don't* enqueue A and D (we've encountered them already), but we do enqueue F. We also add it to the explored set.

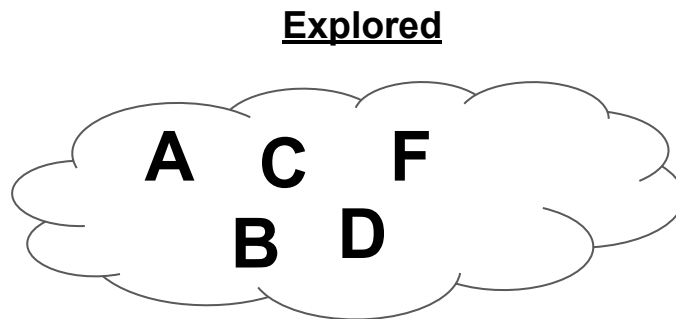
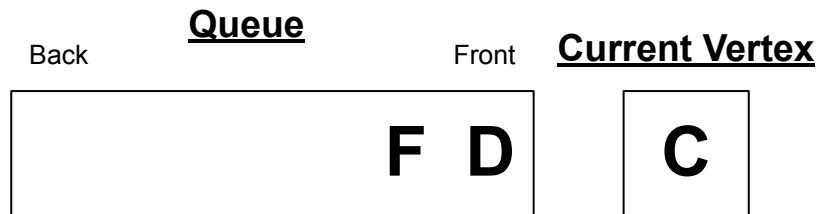
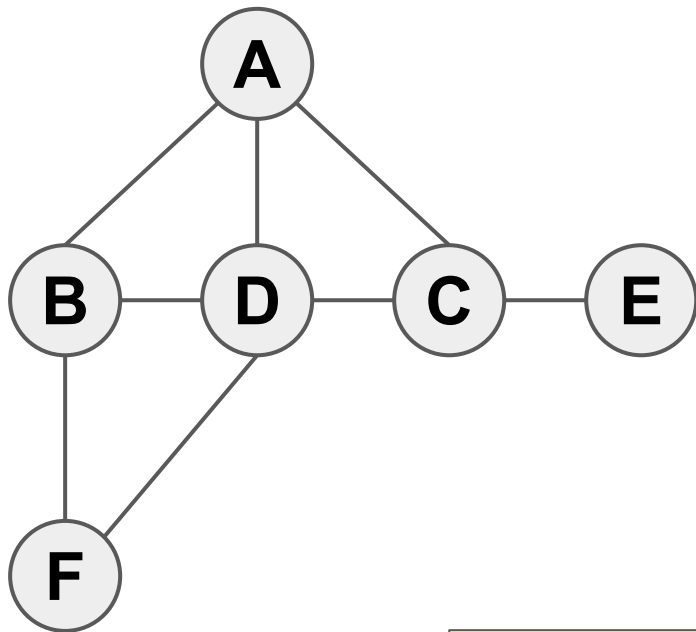


BFS Traversal: A B



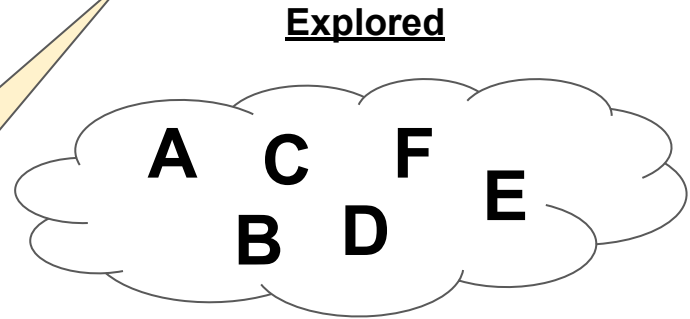
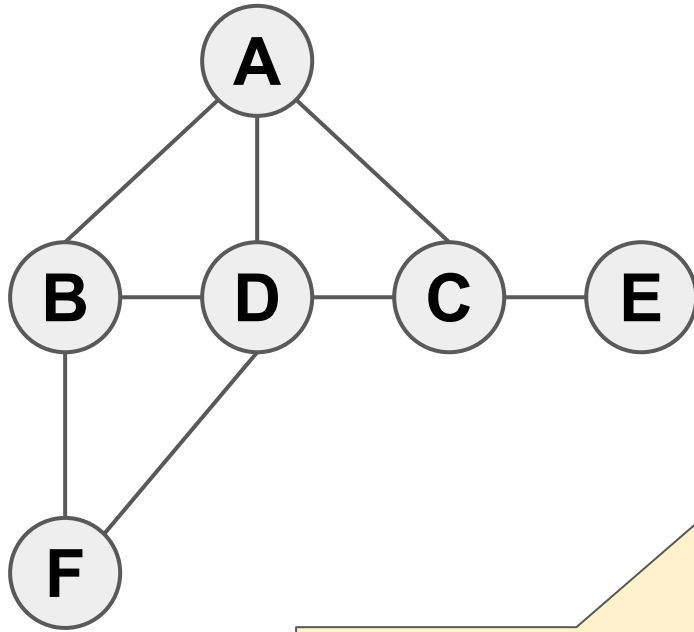
We dequeue C. It is now our current vertex.

BFS Traversal: A B



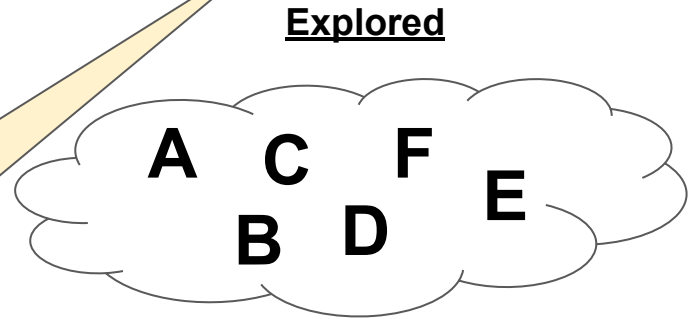
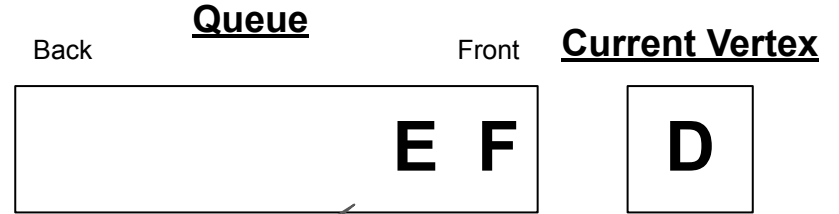
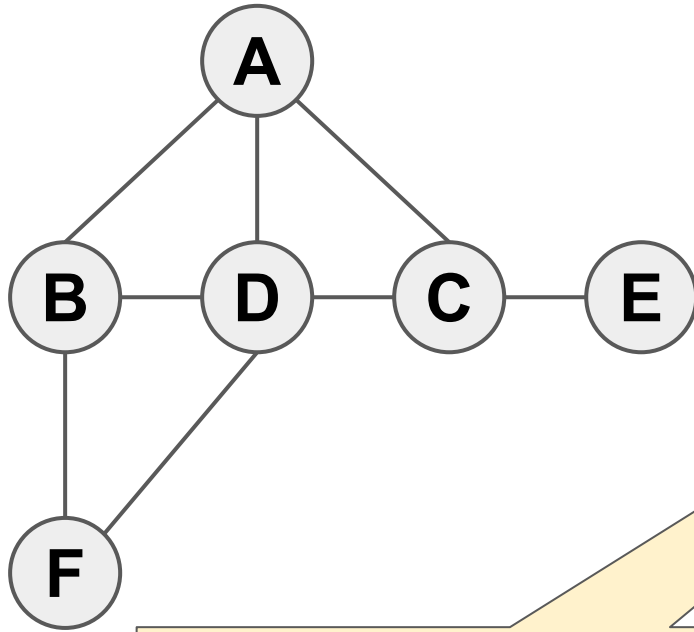
We process C.

BFS Traversal: A B C



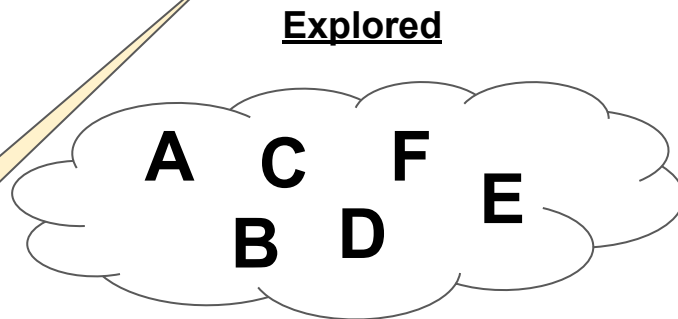
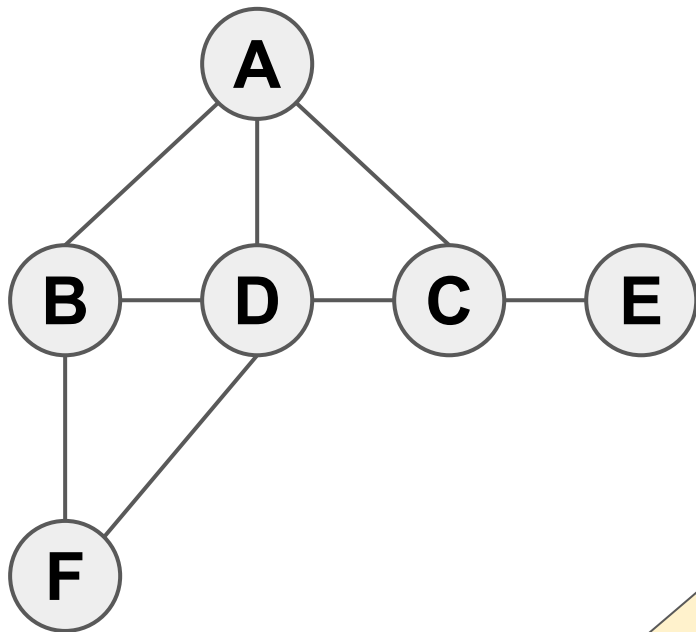
And we enqueue its only unexplored neighbor (E) and add it to the set of explored vertices,

BFS Traversal: A B C



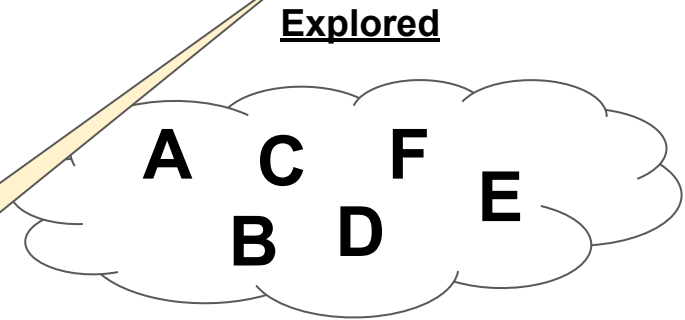
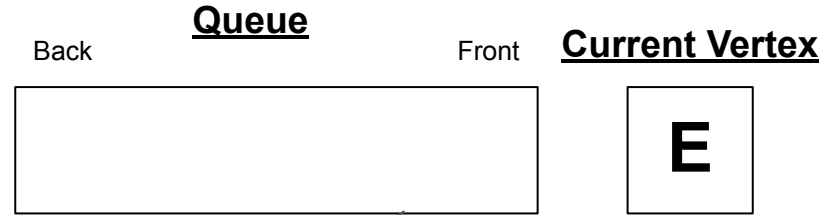
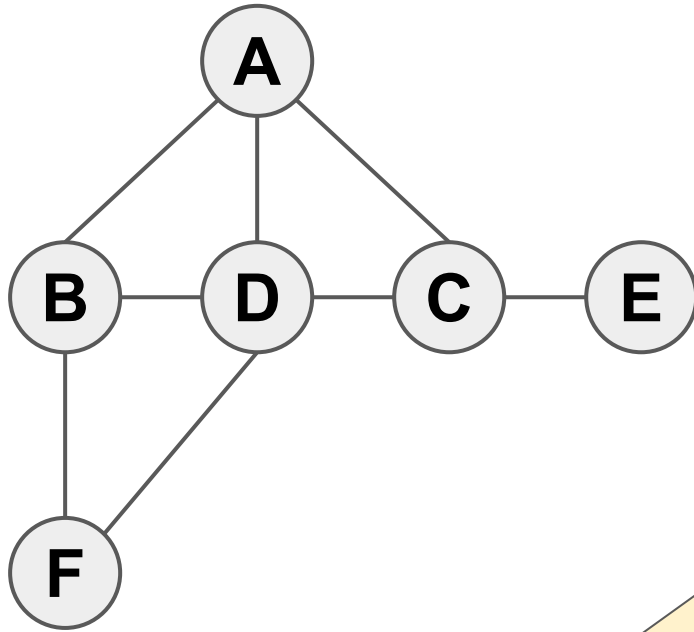
We dequeue D. It becomes our current vertex. We process it, but don't add any more vertices to the queue, because all its neighbors have already been explored.

BFS Traversal: A B C D



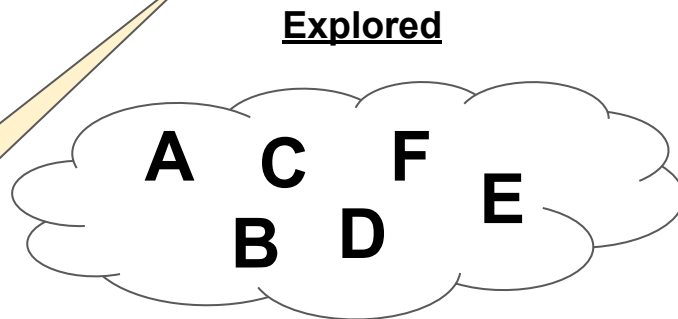
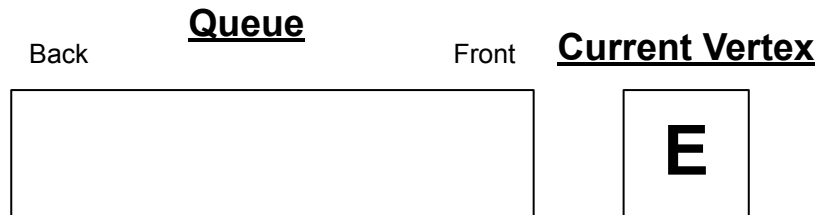
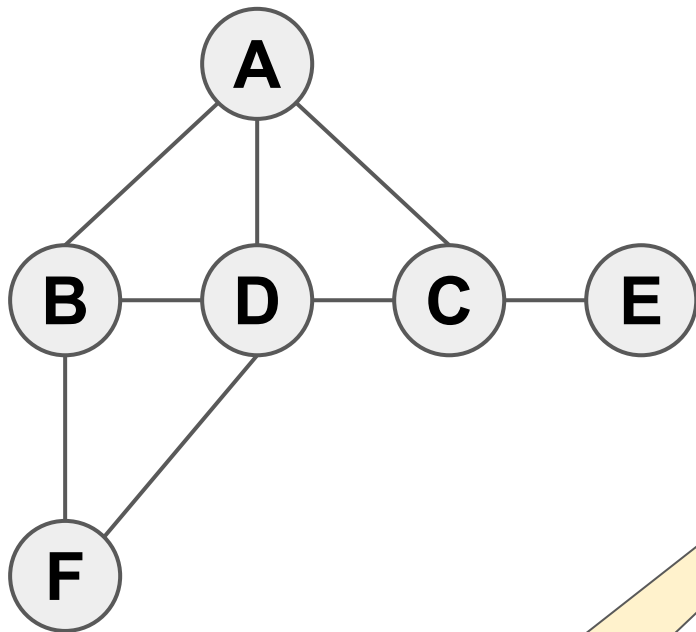
Same for F

BFS Traversal: A B C D F



And same for E

BFS Traversal: A B C D F E



Once the queue is empty, BFS is done.

BFS Traversal: A B C D F E