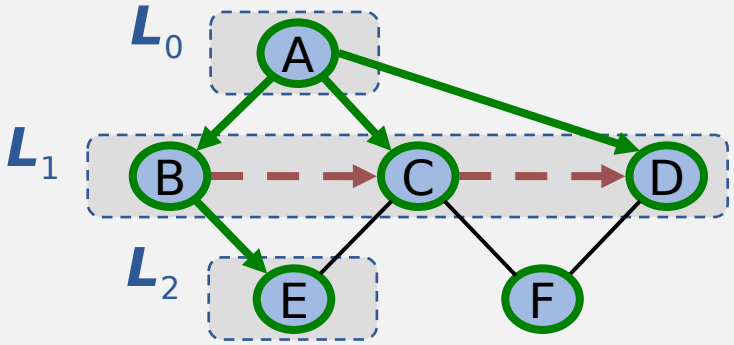
CALIFORNIA STATE UNIVERSITY
FULLERTON

CPSC 131

Data Structures

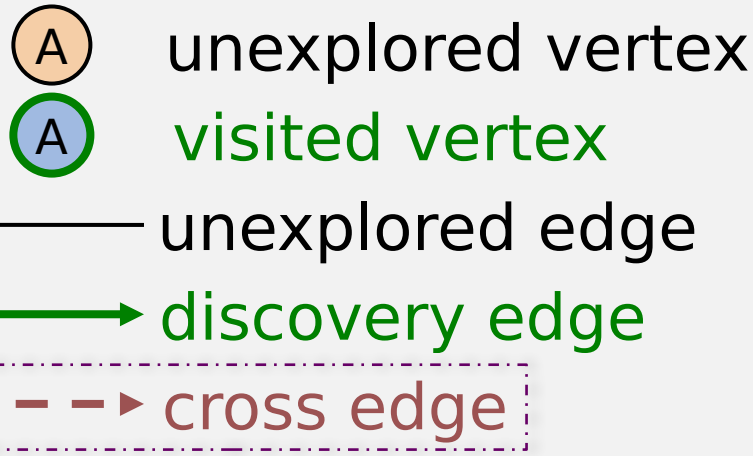


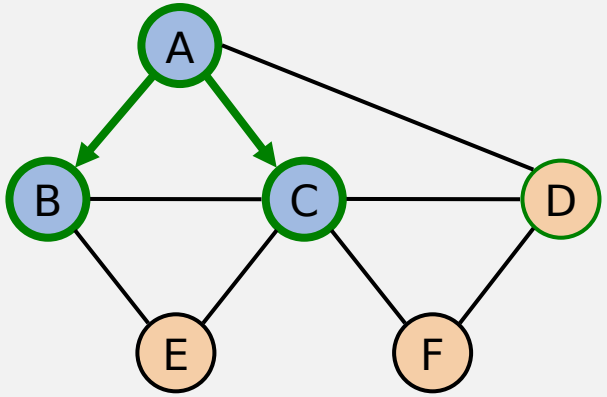
Breadth-First Search (BFS)

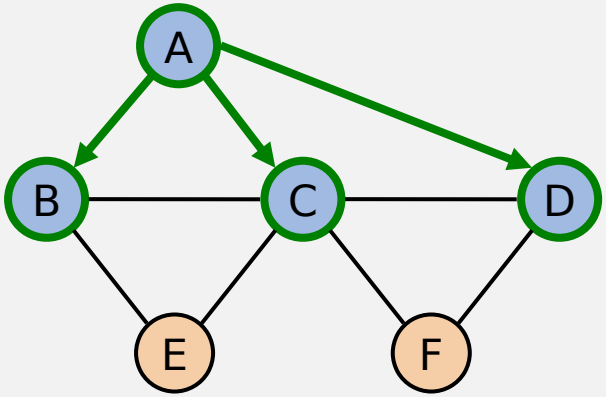
Review: Graph Traversals

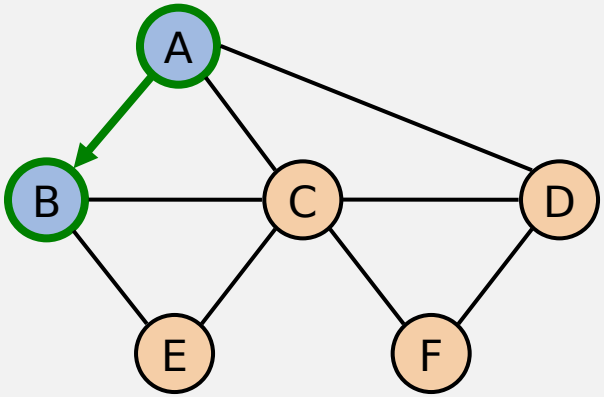
- A systematic procedure for exploring a graph by examining all of its vertices and edges
- Traversal algorithms
  - Depth-First Search (DFS)
    - Visits the child vertices before visiting the sibling vertices
    - A stack OR recursion is used when implementing DFS
  - Breadth-First Search (BFS)
    - Visits the neighbor vertices before visiting the child vertices
    - A queue is used in the search process

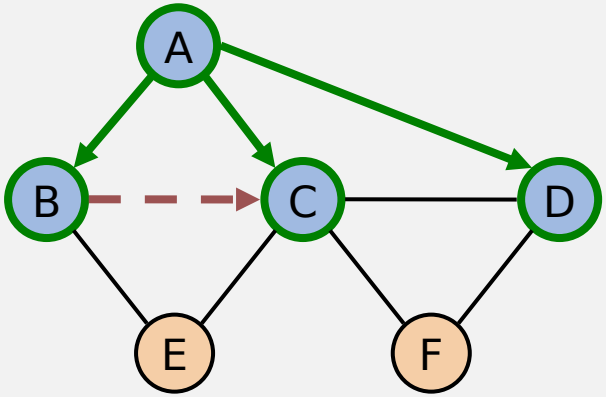
BFS Traversal
Terminologies & Sketches

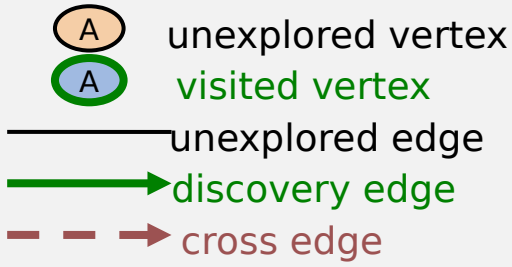




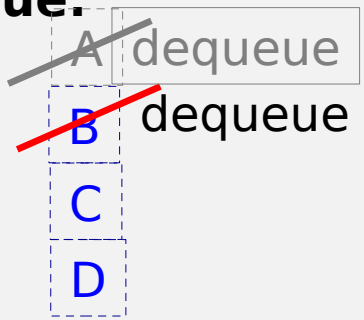


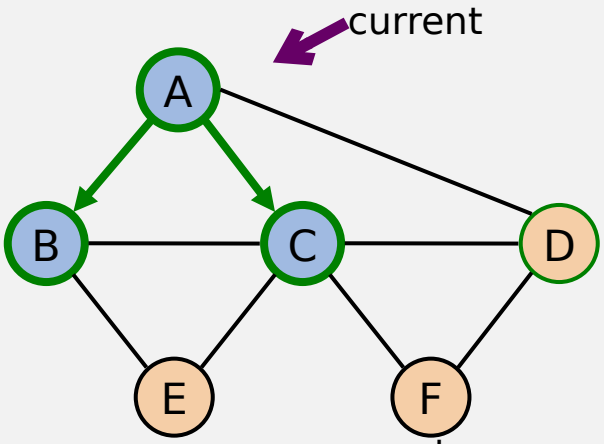


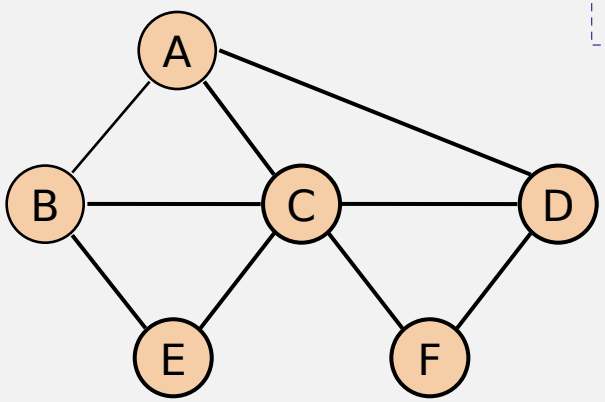


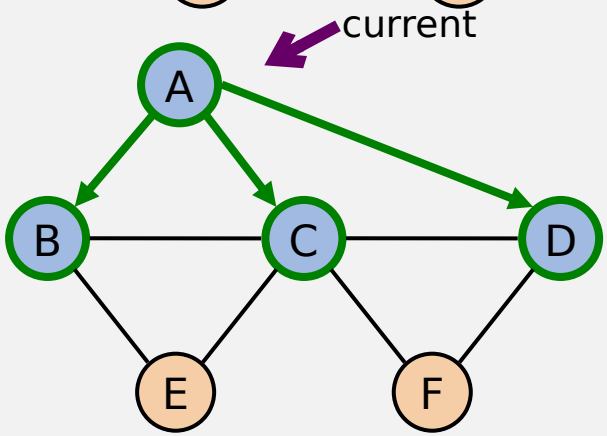

A
unexplored vertex
A
visited vertex
unexplored edge
discovery edge
cross edge

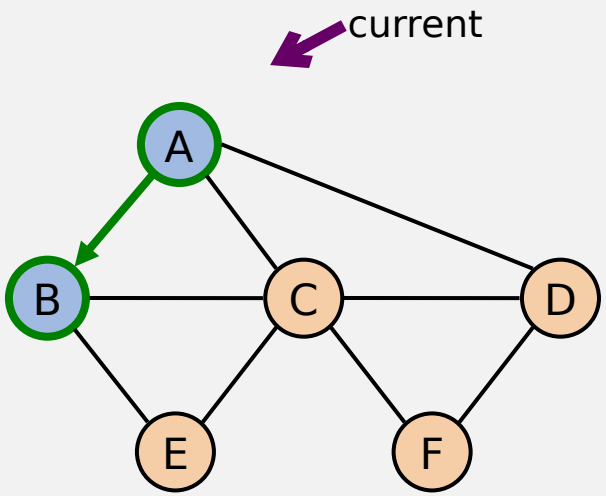
Queue:

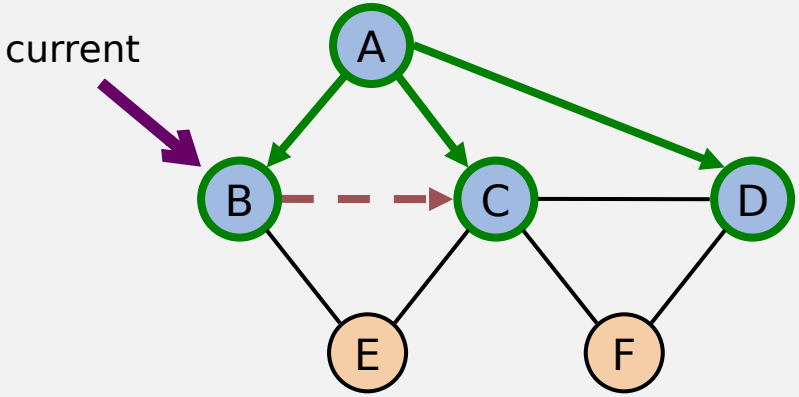

A
dequeue
B
dequeue
C
D

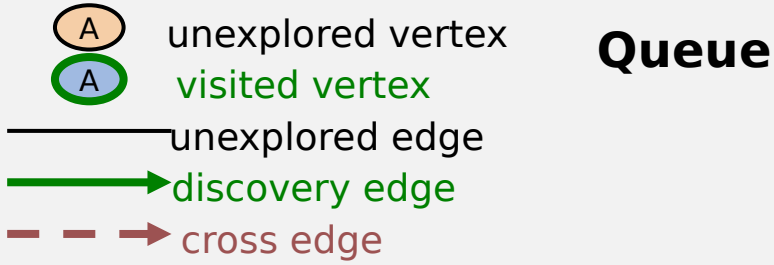

current
A
B
C
D
E
F


A
B
C
D
E
F


current
A
B
C
D
E
F


current
A
B
C
D
E
F


current
A
B
C
D
E
F



Queue:

A dequeue

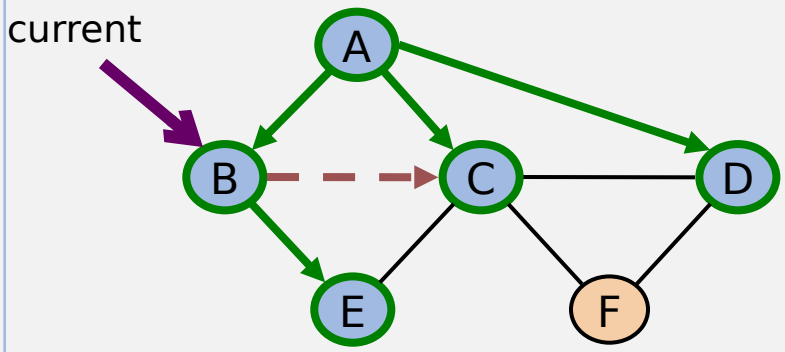
B dequeue

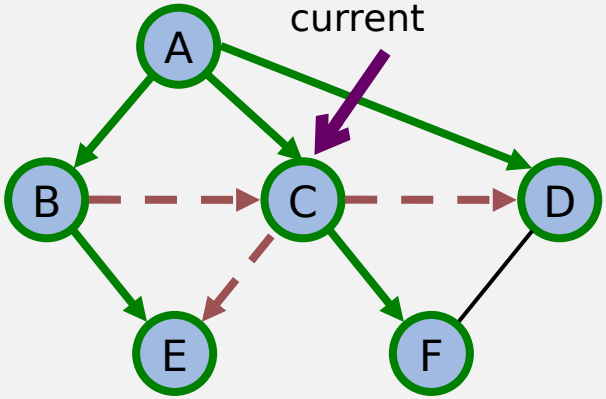
C dequeue

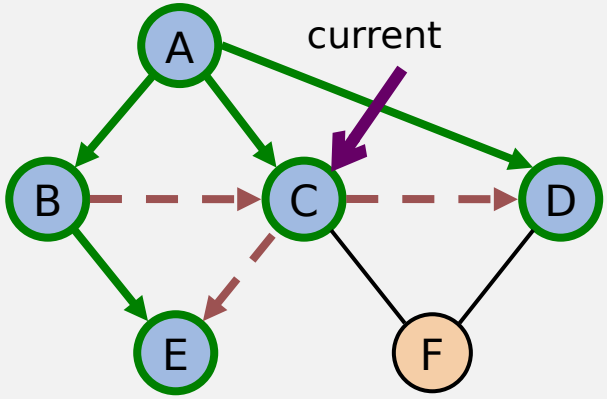
D dequeue

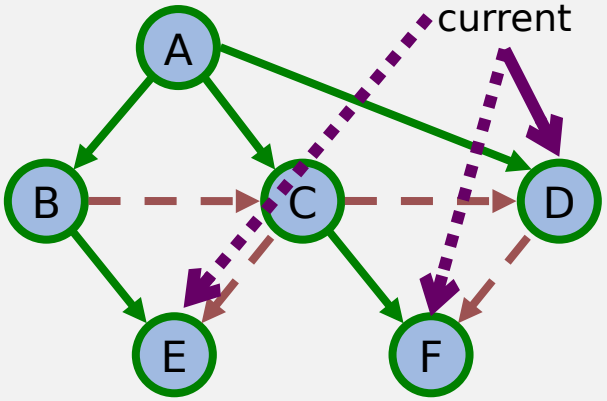
E dequeue

F dequeue









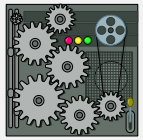
BFS Algorithm Pseudo Code

```
BFS(startV) {
   Set all vertices to not_visited
   Push startV to queue

   Set    startV to visited

   while (queue is not empty )
      currentV = Pop queue
      "Visit" current

      for each vertex adjV adjacent to currentV do
         if ( adjV is not visited)
            Push adjV to queue
            Set   adjV to visited
}
```



Analysis of BFS

- Setting/getting a vertex/edge label takes $O(1)$ time
- Each vertex is labeled twice
  - once as NOT VISITED
  - once as VISITED
- For-loop is called once for each vertex
- BFS runs in $O(n + m)$ time provided the graph with n vertices and m edges is represented by the adjacency list structure

How can one find and report a shortest path using BFS?