

Algorithms

(Graphs)

Pramod Ganapathi

Department of Computer Science
State University of New York at Stony Brook

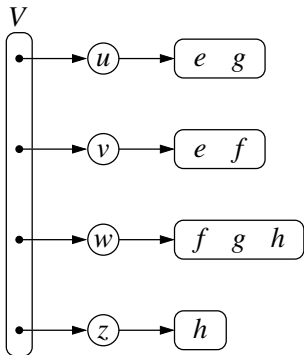
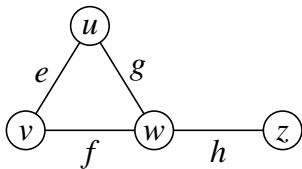
March 25, 2021



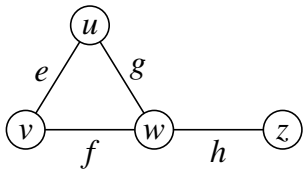
Graph representations

- Adjacency list
- Adjacency matrix

Adjacency list



Adjacency matrix

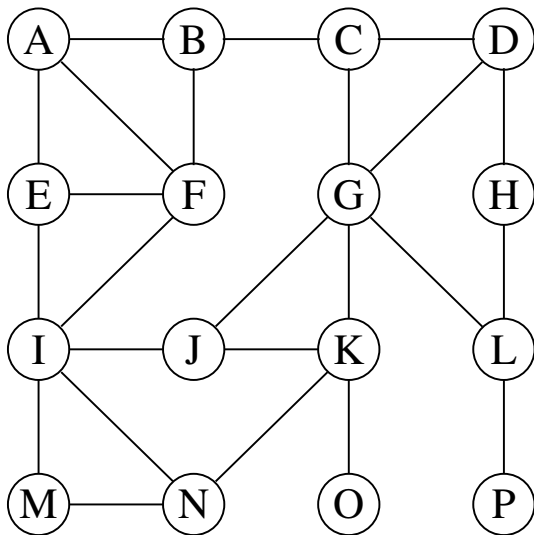


	0	1	2	3
$u \rightarrow$	0	e	g	
$v \rightarrow$	1	e	f	
$w \rightarrow$	2	g	f	h
$z \rightarrow$	3		h	

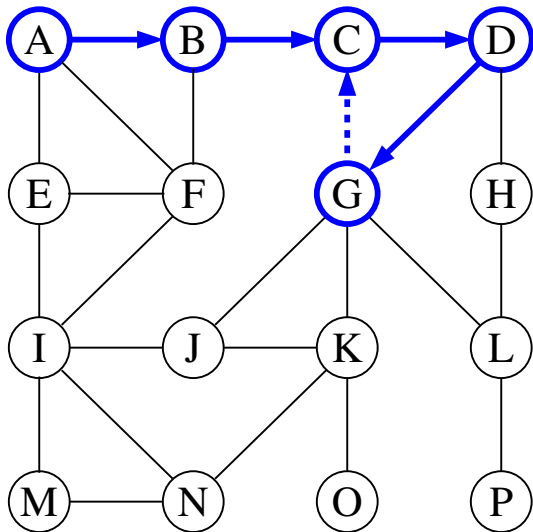
Graph traversals

- Depth first search (DFS)
- Breadth first search (BFS)

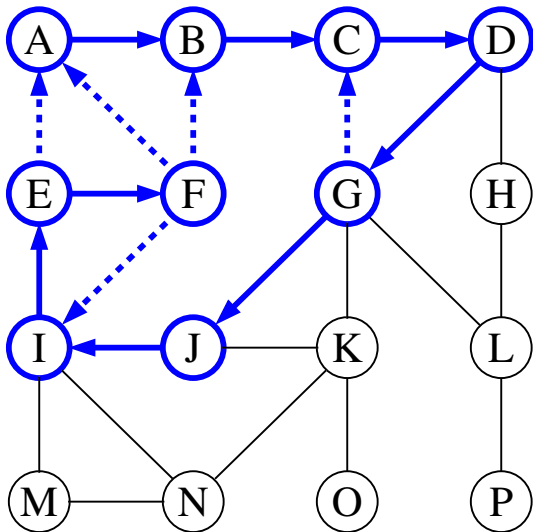
DFS



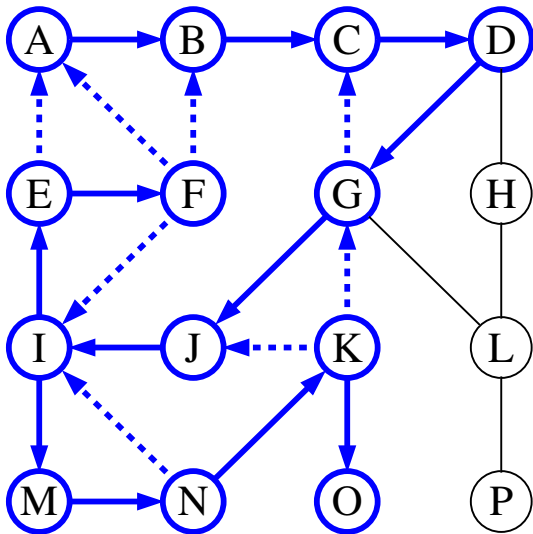
DFS

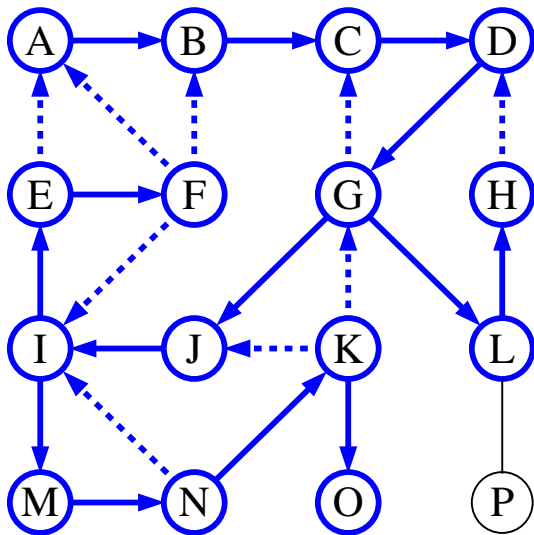


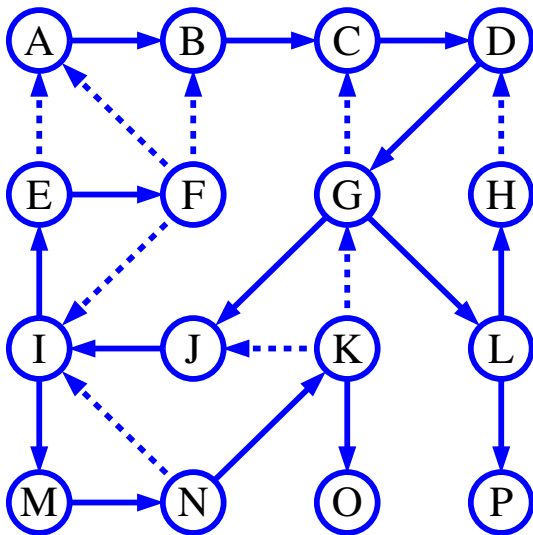
DFS



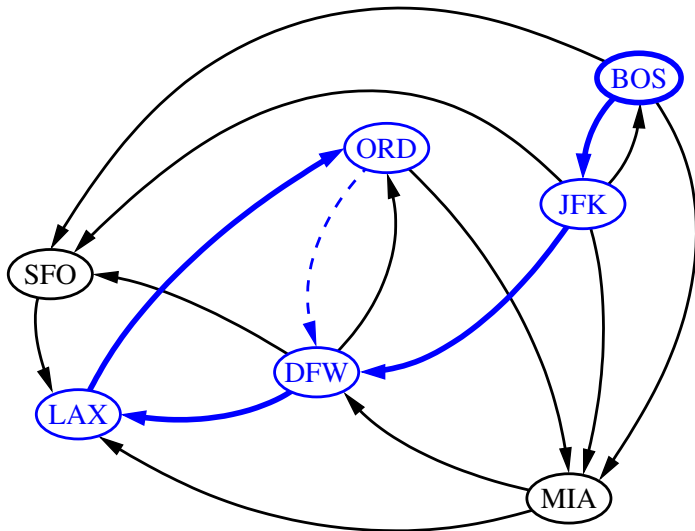
DFS



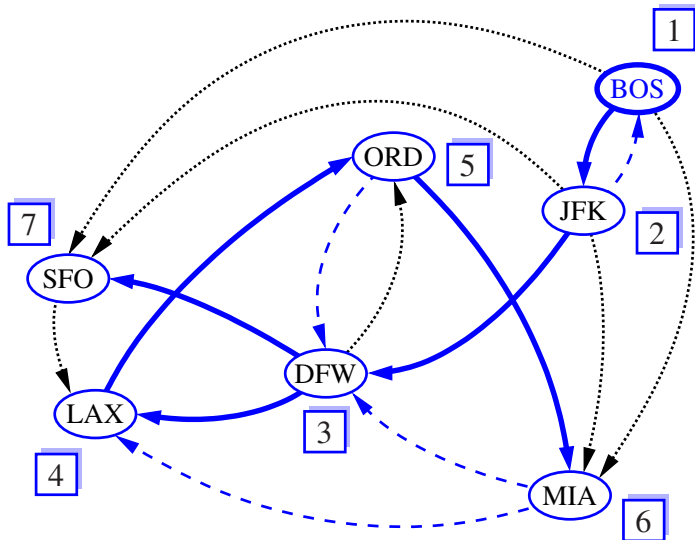




DFS



DFS



DFS

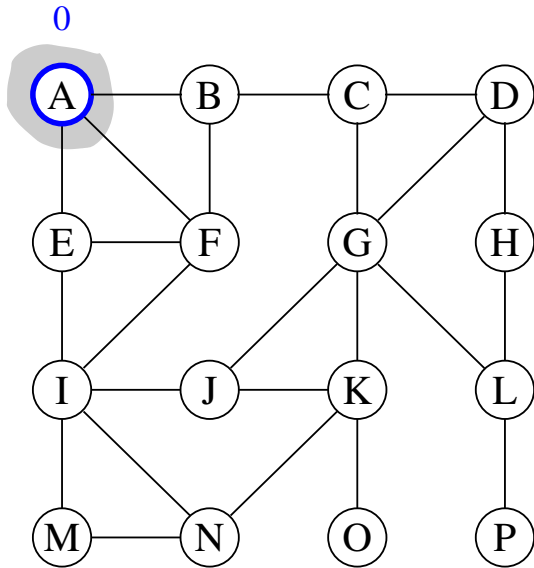
DEPTHFIRSTSEARCH(G)

1. Mark each vertex in V with 0 as a mark of being unvisited
2. $count \leftarrow 0$
3. **for** each vertex v in V **do**
4. **if** v is marked with 0 **then**
5. DFS(v)

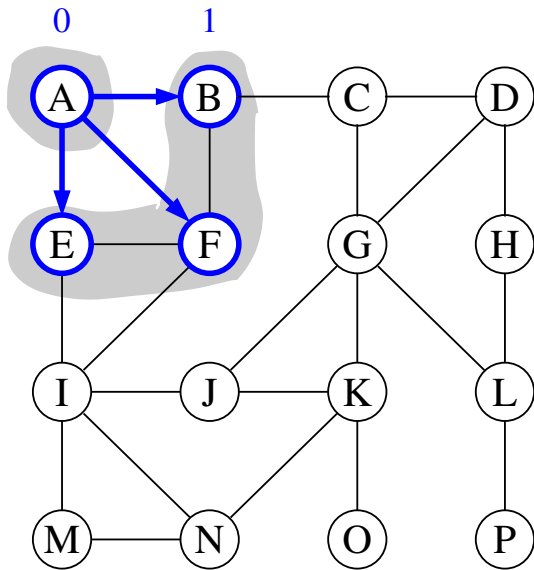
DFS(v)

1. $count \leftarrow count + 1$
2. Mark v with $count$
3. **for** each vertex w in V adjacent to v **do**
4. **if** w is marked with 0 **then**
5. DFS(w)

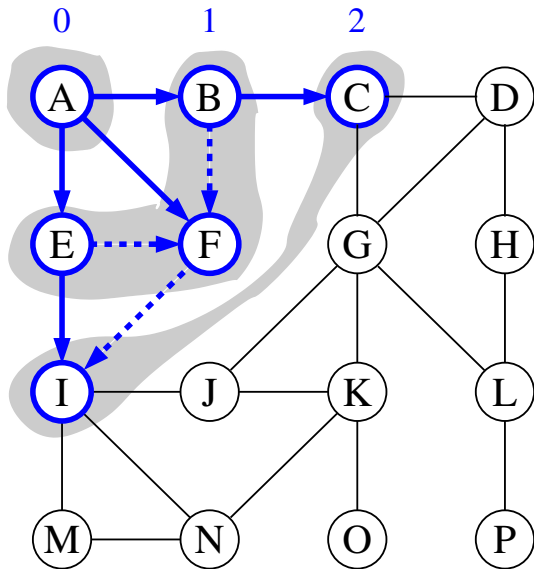
BFS



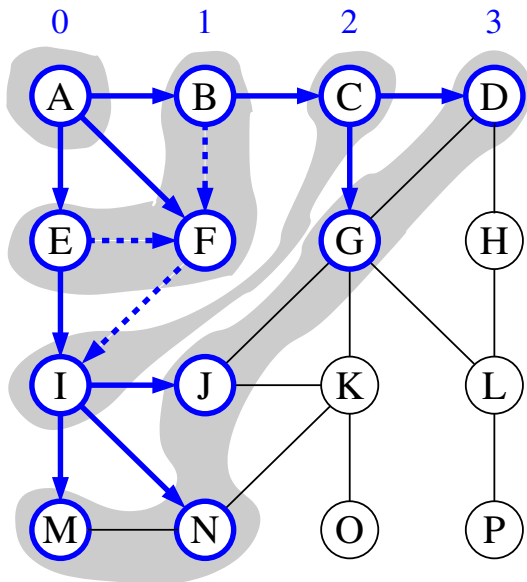
BFS



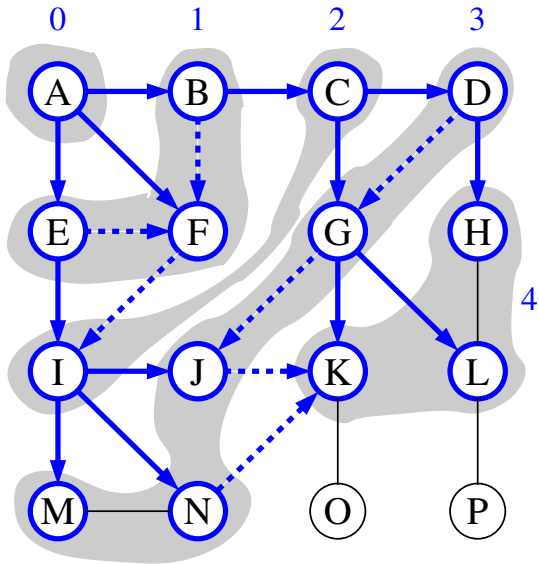
BFS



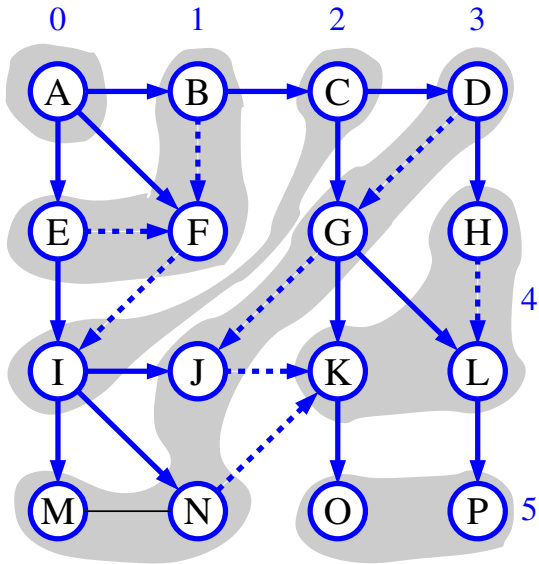
BFS



BFS



BFS



BFS

BREADTHFIRSTSEARCH(G)

1. Mark each vertex in V with 0 as a mark of being unvisited
2. $count \leftarrow 0$
3. **for** each vertex v in V **do**
4. **if** v is marked with 0 **then**
5. BFS(v)

BFS(v)

1. $count \leftarrow count + 1$
2. Mark v with $count$
3. Initialize a queue with v
4. **while** queue is not empty **do**
5. **for** each vertex w in V adjacent to the front vertex **do**
6. **if** w is marked with 0 **then**
7. $count \leftarrow count + 1$
8. Mark w with $count$
9. Add w to the queue
10. Remove the front vertex from the queue