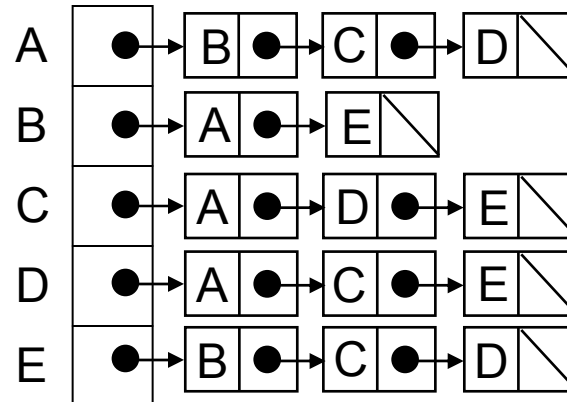
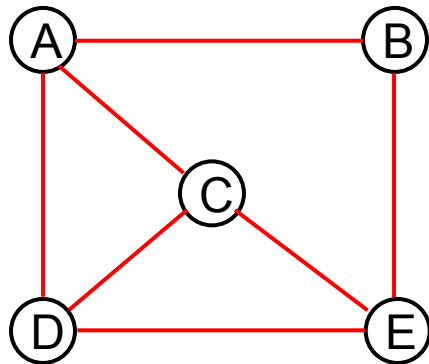


# Adjacency Lists

- Adjacency Lists
  - Adjacency list of a vertex  $v$  = sequence of vertices adjacent to  $v$
  - Represent the graph by the adjacency lists of all the vertices



# Breadth First Search (BFS)

mark all vertices as not visited.

s = starting vertex

put s in the queue

while the queue is not empty {

    current node = dequeue()

    for each neighbor of the current node {

        if the neighbor is not marked {

            visit and mark neighbor

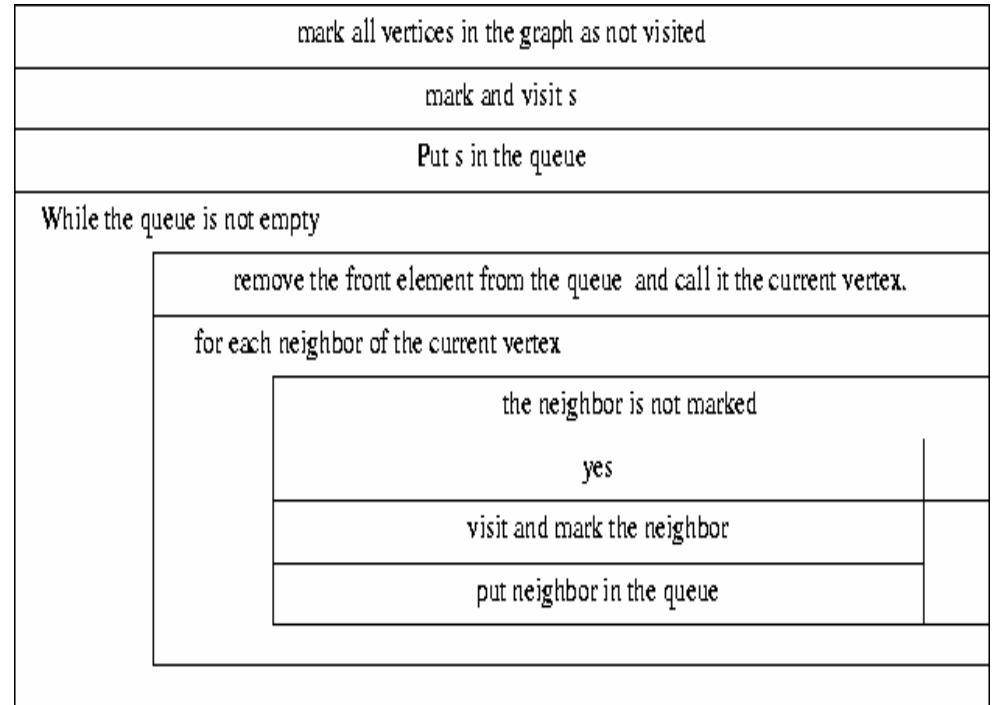
            enqueue the neighbor

            display current node, neighbor

        }

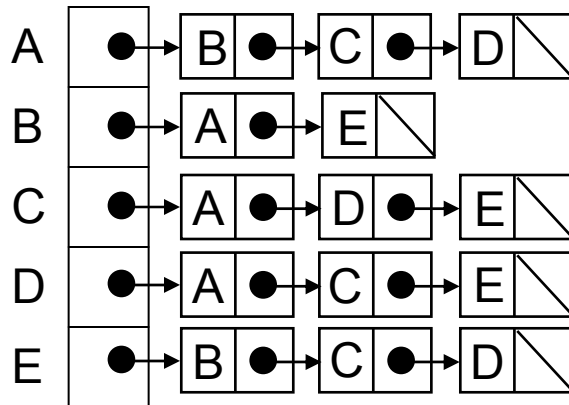
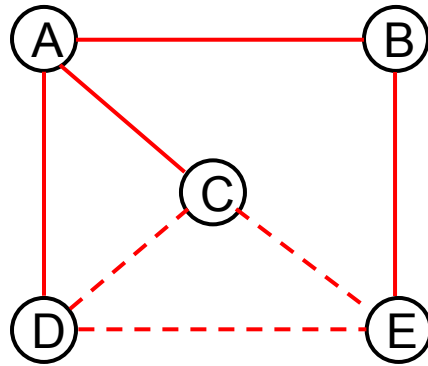
    }

}



Animated BFS and DFS :

<http://www.informatik.uni-bonn.de/~jens/netdays/english/searching/index.html>



Display

A,B
A,C
A,D
B,E

## BFS Example (using adjacency list)

all vertices marked not visited

start vertex = A

enqueue(A) onto queue Q

Q = { A }

current vertex = dequeue() -> A

for each neighbor of A {

is B marked ? No - mark B, enqueue(B), display A,B

is C marked ? No - mark C, enqueue(C), display A,C

is D marked ? No - mark D, enqueue(D), display A,D

}

Q = { B, C, D }

current vertex = dequeue() -> B

for each neighbor of B {

is A marked ? Yes

is E marked ? No - mark E, enqueue(E), display B,E

}

Q = { C, D, E }

current vertex = dequeue() -> C

for each neighbor of C -> A,D,E all marked

Q = { D, E }

current vertex = dequeue() -> D

for each neighbor of D -> A,C,E all marked

Q = { E }

current vertex = dequeue() -> E

for each neighbor of E -> B,C,D all marked

Q = {empty}

done.