

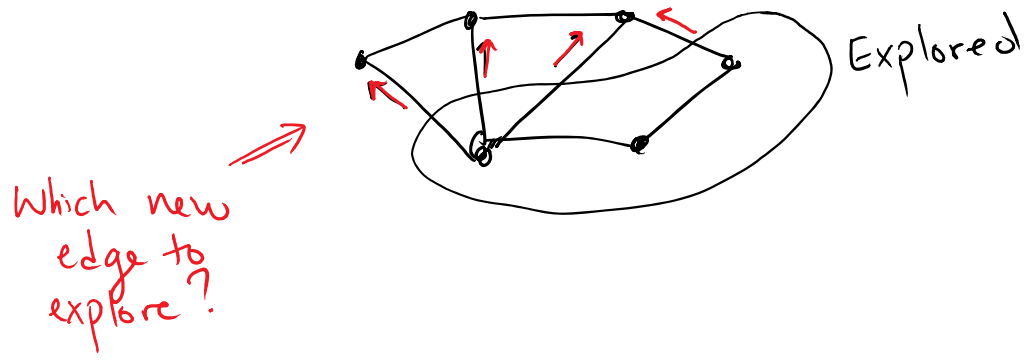
# Breadth-First-Search (BFS)

## Generic Search Alg:

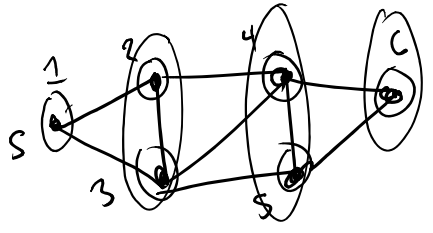
1.  $Exp = \{s\}$        $Exp =$  set of explored nodes
2. While  $(\exists \{u,v\} \in E : (u \in Exp \wedge v \notin Exp))$
3.     Add  $v$  to  $Exp$

## Big Question:

If multiple edges cross boundary between explored and unexplored, which do explore first?



One strategy: explore all edges crossing current boundary, then look at new boundary & explore



← Breadth-First Search  
vs  
Depth-First Search

Idea, explore layers.

$Exp[v] = 0 \quad \forall v \in V$  // mark True when "explored"

$A = \{\}$   
 $A.add(s)$

$Exp[s] = True$

A is a queue = "First in, first out"

while (A is not empty) {

$v = A.pop$

For each edge  $(v, w)$  {

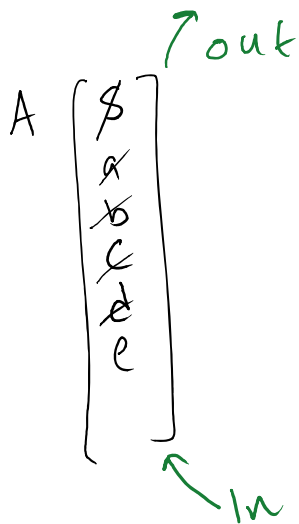
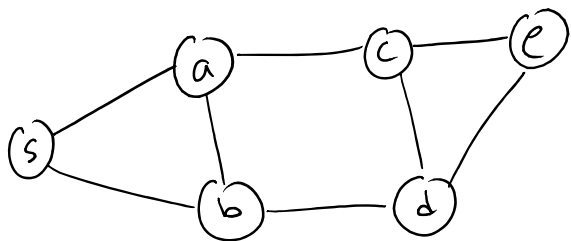
If  $(Exp[w] = false)$  {  $Exp[w] = True; A.add(w);$  }

}

}

This is Breadth First Search - slowly move away in layers from initial node

ex:



exp

s	T
a	F
b	F
c	F
d	F
e	F

Q:

What is runtime of BFS using adjacency list if  $n$  is total # vertices,  $m$  is total # edges,  $n_s$  is # of vertices reachable from  $s$ ,  $m_s$  is # edges reachable from  $s$ .

A.  $O(m_s)$

B.  $O(n+m_s)$

C.  $O(n_s \cdot m_s)$

D.  $O(n+n_s m_s)$

Answer: B.

1: initializing  $Exp[v] \forall v \in V$  takes time  $O(n)$