

$$G = (V, E) \rightarrow \{(a,b), (a,d), (d,c), (e,b), (d,e), (c,e)\}$$

$$\hookrightarrow \{a, b, c, d, e\}$$

storage: 1) adjacency matrix

1 row, 1 col for each vertex

$$a_{ij} = \begin{cases} 1 & \text{if edge } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

can check if  $(u, v) \in E$  in  $O(1)$  time

can iterate through neighbors of  $u$  in  $O(V)$

can iterate through all edges in  $O(V^2)$

$\Theta(V^2)$  storage space

2) adjacency list

for each vertex  $u$ , maintain a linked list  
s.t.  $v$  is on list iff  $(u, v) \in E$

for each  $v \in V$

for each  $u \in \text{Adj}[v]$

process  $(v, u)$

check  $(u, v) \in E$  in  $O(V)$  time

iterate through all edges in  $O(E+V)$  time

1



## BFS (v)

let  $Q = \emptyset$   
 $Q.enqueue(v)$   
 $d[v] \leftarrow 0$   
 $\forall u \in V, color[u] \leftarrow WHITE$   
 $color[v] = GRAY$

color is BLACK if done  
GRAY if seen but not explored  
(in the queue)  
WHITE not seen

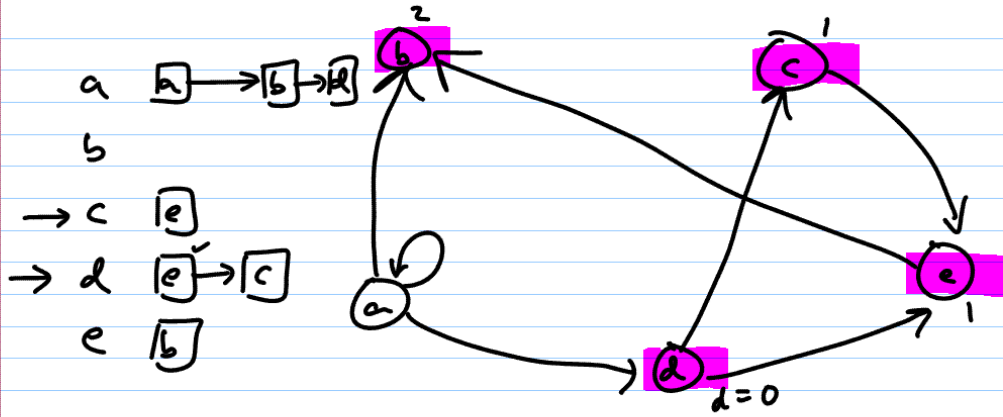
while ( $Q \neq \emptyset$ )  
     $u \leftarrow Q.dequeue()$   
    for each  $w \in Adj[u]$   
        if ( $color[w] = WHITE$ )  
             $Q.enqueue(w)$   
             $color[w] = GRAY$   
             $d[w] = d[u] + 1$   
     $color[u] = BLACK$

executes at most once  
for each vertex

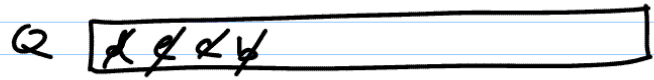
executes at most once  
for each edge

$O(V + E)$   
or  
 $O(E)$

(considered linear for  
graph algs)



gray  
black



BFS(d)