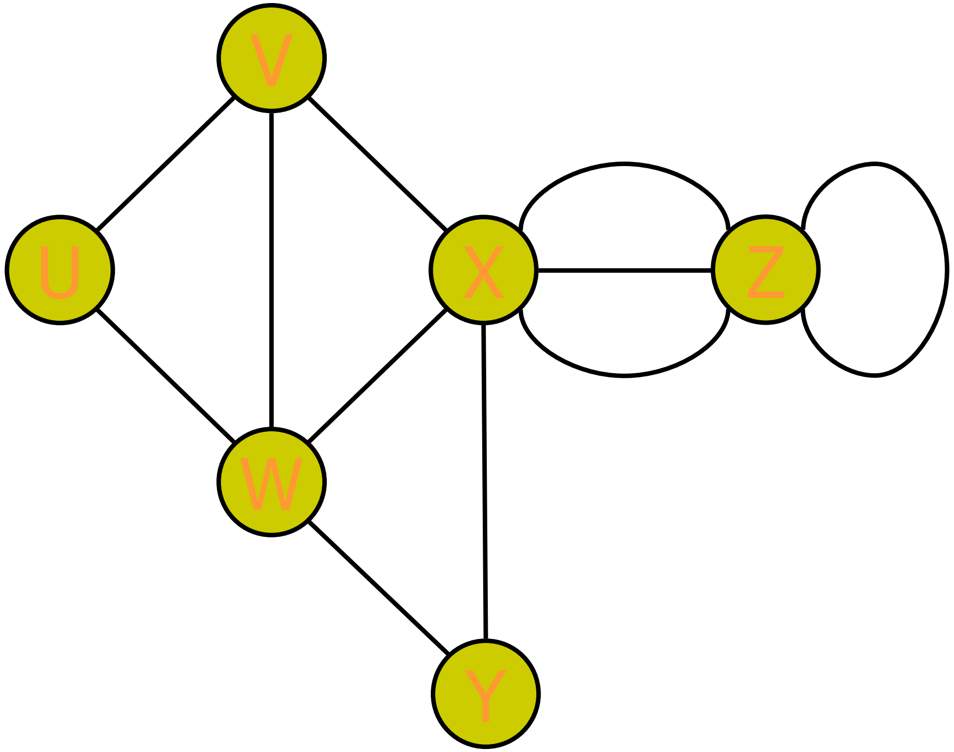
Algorithms

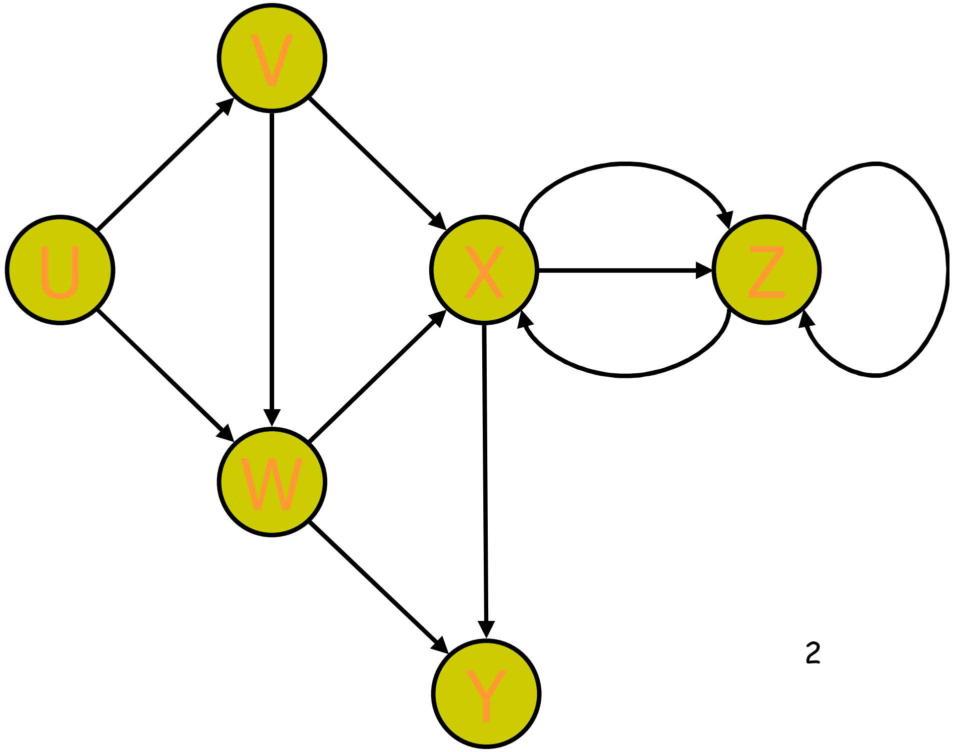
Lesson #5:
BFS

(Based on slides by Prof. Dana Shapira)

Graph

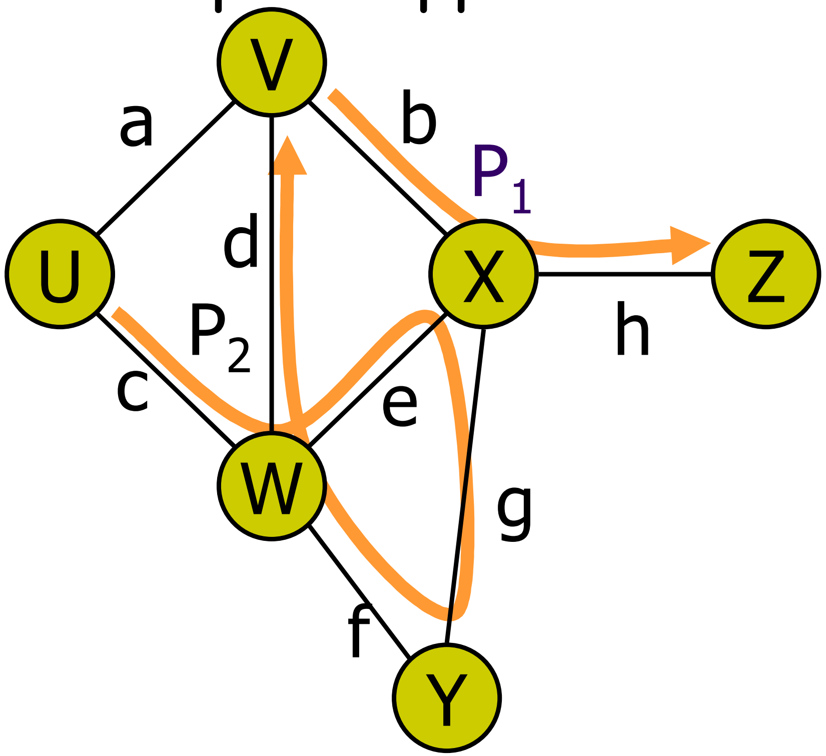
- A graph is a pair (***V***, ***E***), where
  - ***V*** is a set of nodes, called vertices
  - ***E*** is a collection of pairs of vertices, called edges $\subseteq (V \times V)$
- If edge pairs are ordered, the graph is *directed*, otherwise *undirected*.





Paths

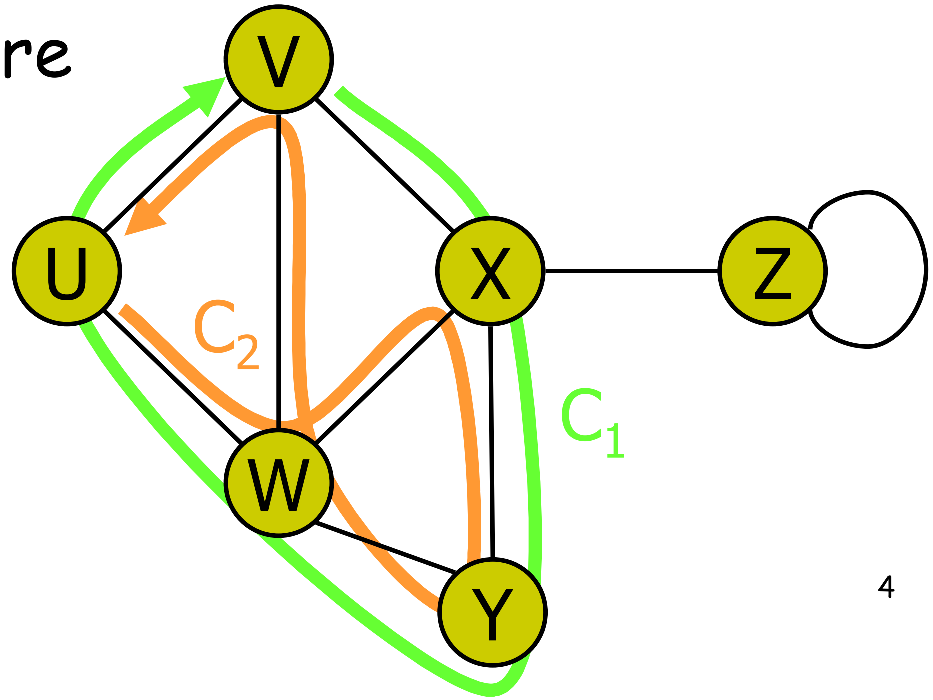
- **Path**
  - sequence of vertices $\{v_0, v_1, \ldots, v_p\}$ where $(v_i, v_{i+1}) \in E$
- **Simple path**
  - If no vertex in the path appears more than once



Cycles

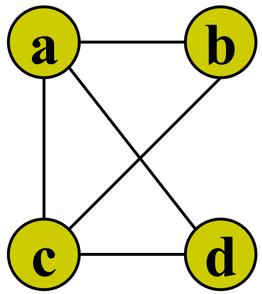
- **Circuit**
  - A path $\{v_0, v_1, \ldots, v_p\}$ where $v_0 = v_p$.
- **Simple circuit**
  - If no vertex, other than the start-end vertex, appears more than once, and the start-end vertex does not appear elsewhere

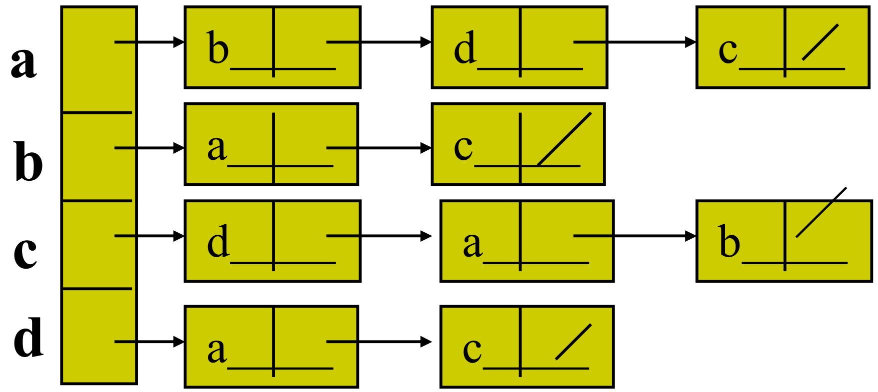
    in the circuit



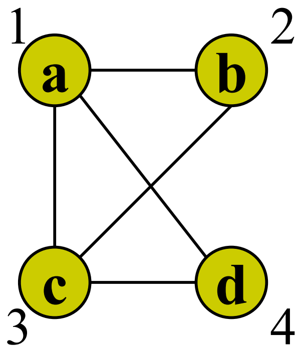
Graph Representations

- *Adjacency Lists.*





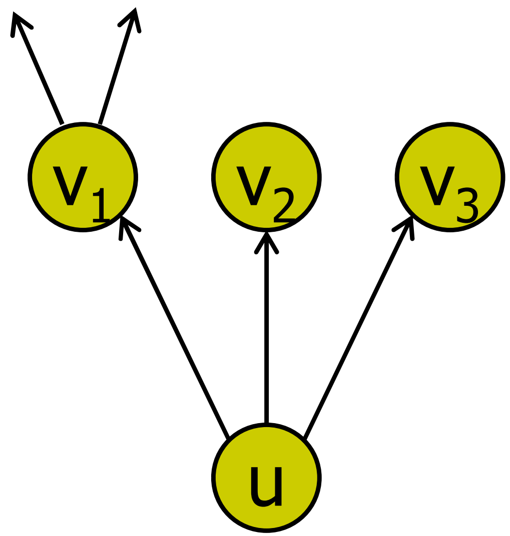
- *Adjacency Matrix.*



	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0

Breadth-First Search & Depth-First Search

Objective: Visit all the vertices of the graph, by following the edges



For each node u:

BFS: Visit all the nodes reachable from u before continuing with nodes reachable from v1

DFS: Visit all the nodes reachable from v1 before continuing with v2

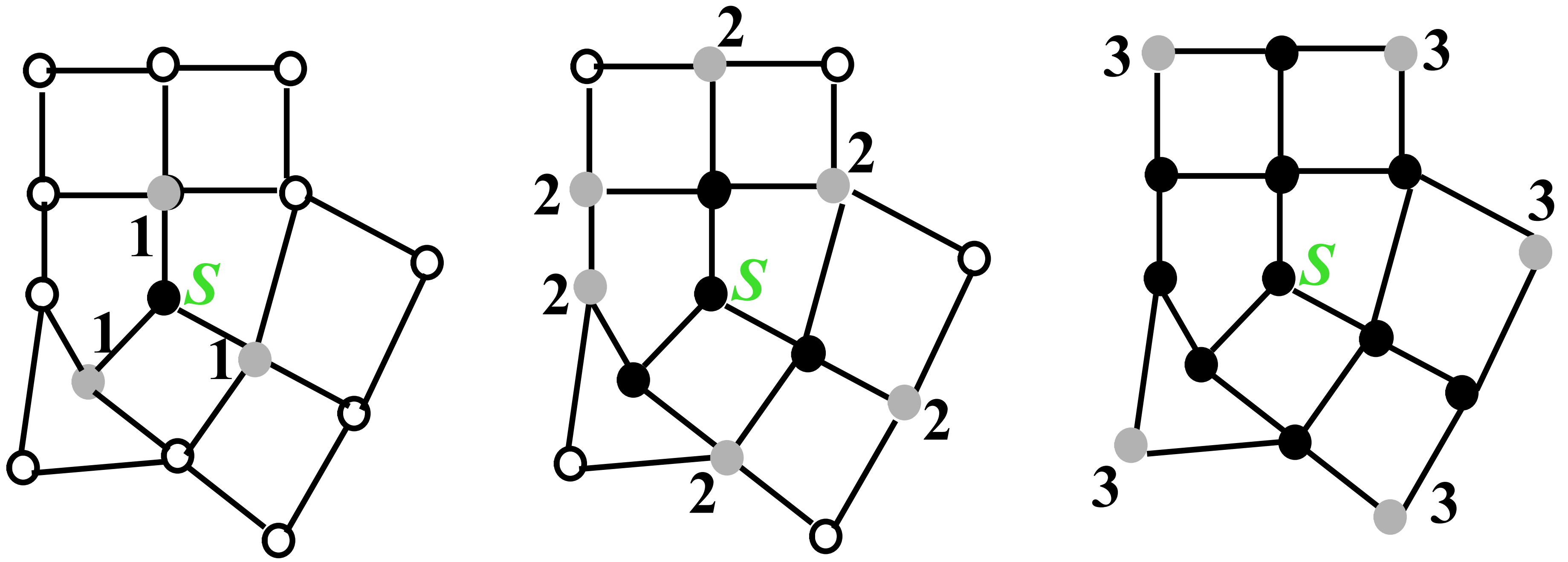
Keep some info about the search process – will be useful for applications

Breadth-first Search

- **Input:** Graph $G = (V, E)$, directed or undirected, and ***source vertex*** $s \in V$.
- **Output:**
  for all $v \in V$
  - $d[v]$ = distance from s to v.
  - $\pi[v] = u$ such that (u, v) is the last edge on shortest path from s *to* v.
  - Builds ***breadth-first tree*** with root s that contains all reachable vertices.

- Colors the vertices to keep track of progress.
  - *White* – Undiscovered.
  - *Gray* – Discovered but not finished.
  - ***Black*** – Finished.

BFS for Shortest Paths



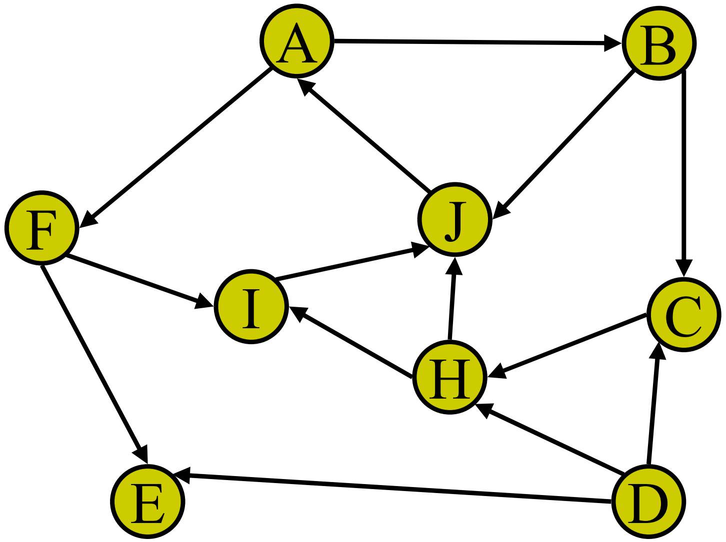
● Finished ● Discovered ○ Undiscovered

```
BFS(G,s)
for each vertex u in V[G] - {s}
        do color[u] ← white
           d[u] ← ∞
           π[u] ← NULL
color[s] ← gray
d[s] ← 0
π[s] ← NULL
Q ← Φ
enqueue(Q,s)
while Q ≠ Φ
        do u ← dequeue(Q)
                for each v in Adj[u]
                        do if color[v] = white
                                then color[v] ← gray
                                     d[v] ← d[u] + 1
                                     π[v] ← u
                                     enqueue(Q,v)
                color[u] ← black
```

Running time is $O(|V|+|E|)$

Example

Let G be the following graph:



What is BFS(G,A)?

BFS(G,D)?

BFS correctness

Let $G = (V,E)$ be a graph and $s \in V$.

- ***δ(s,v)*** - the *shortest path distance* from s to v (the minimum number of edges).
  - $\delta(s,v) = \infty$ if there is no path from s to v.

Theorem:

1. During its execution, BFS discovers every vertex $v \in V$ that is reachable from s.
2. Upon termination, $d[v] = \delta(s,v)$.
3. For all reachable vertices v except for s, one of the shortest paths from s to v is a shortest path from s to $\pi[v]$ followed by the edge $(\pi[v],v)$.

BFS correctness

Lemma 1: $\forall$ edge $(u,v)\in E$, $\delta(s,v) \le \delta(s,u) + 1$

Lemma 2: Upon termination of BFS: $\forall v \in V$, $d[v] \ge \delta(s,v)$.

BFS correctness

- **Lemma 3:** If the queue Q contains vertices $<v_1, .. v_r>$ where v_1 and v_r are Q's head and tail, then:
  1. $d[v_r] \le d[v_1] + 1$
  2. $d[v_i] \le d[v_{i+1}]$ for i=1,2,..,r–1

- **Corollary :** If vertex v_i was enqueued before vertex v_j during BFS, then $d[v_i] \le d[v_j]$ when v_j is enqueued.

Proof of the Theorem

- By contradiction, assume that there is a vertex v such that $d[v] \neq \delta(s,v)$, and choose v with minimum $\delta(s,v)$. Using Lemma 2 $d[v] > \delta(s,v)$.
  - $v \neq s$, since $d[s]=0=\delta(s,s)$.
  - v is reachable from s for otherwise $\delta(s,v)=\infty \geq d[v]$.

  Let u be the vertex immediately preceding v on a shortest path from s to v.
  - $\delta(s,v) = \delta(s,u)+1. \Rightarrow \delta(s,u)<\delta(s,v) \Rightarrow$ from minimality $\delta(s,v)$, $d[u] = \delta(s,u)$.

  * $d[v]>\delta(s,v)=\delta(s,u)+1=d[u] + 1$

Proof (cont.)

Consider now the time when u is dequeued. Vertex v is either white, gray, or black.

- v is white: then in line 15 $d[v]=d[u]+1$ which contradicts *.
- v is gray: There exists w$\in$V that was dequeued before u $\Rightarrow$

  $d[v]=d[w]+1$. but $d[w]\leq d[u]$ (corollary) $\Rightarrow$ $d[v]\leq d[u]+1$ which contradicts *.
- v is black: v was already dequeued $\Rightarrow$ $d[v] \leq d[u]$ (corollary) thus $d[v]<d[u]+1$ which contradicts *.

$\Rightarrow d[v] = \delta(s,v)$

- If $\pi[v]=u$, then $d[v]=d[u]+1$. Thus, we obtain the shortest path from s to v by taking a shortest path from s to $\pi[v]$ and then traversing the edge $(\pi[v],v)$.

Breadth-first Tree

- For a graph $G = (V, E)$ with source s, the ***predecessor subgraph*** of G is $G_\pi = (V_\pi, E_\pi)$ where
  - $V_\pi = \{v \in V : \pi[v] \neq \text{NULL}\} \cup \{s\}$
  - $E_\pi = \{(\pi[v], v) \in E : v \in V_\pi - \{s\}\}$
- The predecessor subgraph G_π is a ***breadth-first tree*** if:
  - V_π consists of the vertices reachable from s and
  - for all $v \in V_\pi$, there is a unique simple path from s to v in G_π that is also a shortest path from s to v in G.
- The edges in E_π are called ***tree edges***.
  $|E_\pi| = |V_\pi| - 1$.

Breadth-first Tree

After performing BFS we can print the vertices on the shortest path from s to v in linear time:

```
print_path(G,s,v)
    if v=s print s
    else if π[v] =NULL
            print ("no path")
            else
                    print_path(G,s, π[v] )
                    print(v)
```