Graphs

- Graph $G = (V, E)$ has vertices (nodes) $V$ and edges (arcs) $E$.
- Graph can be directed or undirected
- Graph can represent any situation with objects and pairwise relationships.
Representations

Adjacency Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Representations

Adjacency List

```
1 → 2 → 3 → 4
2 → 1
3 → 4 → 1
4 → 1 → 3
```
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Space</th>
<th>Query Time</th>
<th>All neighbors time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>$O(V^2)$</td>
<td>$O(1)$</td>
<td>$O(V)$</td>
</tr>
<tr>
<td>List</td>
<td>$O(E)$</td>
<td>$O(\text{degree})$</td>
<td>$O(\text{degree})$</td>
</tr>
</tbody>
</table>

- For a simple graph (no double edges) $E \leq V^2 = O(V^2)$
- For a connected graph $E \geq V - 1$
- For a tree $E = V - 1$
Breadth First Search

- Discover vertices in order of distance from the source.
- Works for undirected and directed graphs. Example is for undirected graphs.
Breadth First Search

\(BFS(G, s)\)

1. for each vertex \(u \in V[G] - \{s\}\)
2. \(\quad \text{do color}[u] \leftarrow \text{WHITE}\)
3. \(\quad d[u] \leftarrow \infty\)
4. \(\quad \pi[u] \leftarrow \text{NIL}\)
5. \(\quad \text{color}[s] \leftarrow \text{GRAY}\)
6. \(\quad d[s] \leftarrow 0\)
7. \(\quad \pi[s] \leftarrow \text{NIL}\)
8. \(\quad Q \leftarrow \emptyset\)
9. \(\quad \text{ENQUEUE}(Q, s)\)
10. while \(Q \neq \emptyset\)
11. \(\quad \text{do } u \leftarrow \text{DEQUEUE}(Q)\)
12. \(\quad \text{for each } v \in \text{Adj}[u]\)
13. \(\quad \text{do if } \text{color}[v] = \text{WHITE}\)
14. \(\quad \quad \text{then } \text{color}[v] \leftarrow \text{GRAY}\)
15. \(\quad \quad d[v] \leftarrow d[u] + 1\)
16. \(\quad \quad \pi[v] \leftarrow u\)
17. \(\quad \quad \text{ENQUEUE}(Q, v)\)
18. \(\quad \quad \text{color}[u] \leftarrow \text{BLACK}\)
Running Time:

1  for each $u \in V$
2    do for each $v \in \text{Adj}(v)$
3      do Something $O(1)$ time

Each edge and vertex is processed once:

$O(E + V) = O(E)$
Depth First Search

- More interesting than BFS
- Works for directed and undirected graphs. Example is for directed graphs.
- Time stamp nodes with discovery and finishing times.
- Lifetime: white, $d(v)$, grey, $f(v)$, black
Code

$DFS(G)$

1. for each vertex $u \in V[G]$
2. 
3. 
4. time $\leftarrow 0$
5. for each vertex $u \in V[G]$
6. 
7. then $DFS\text{-}Visit(u)$

$DFS\text{-}Visit(u)$

1. $color[u] \leftarrow \text{GRAY}$ \hspace{1cm} $\triangleright$ White vertex $u$ has just been discovered.
2. time $\leftarrow time +1$
3. $d[u] \leftarrow time$
4. for each $v \in Adj[u]$ \hspace{1cm} $\triangleright$ Explore edge $(u, v)$. 
5. 
6. 
7. then $\pi[v] \leftarrow u$
8. $DFS\text{-}Visit(v)$
9. $color[u] \leftarrow \text{BLACK}$ \hspace{1cm} $\triangleright$ Blacken $u$; it is finished.
10. $f[u] \leftarrow time \leftarrow time +1$
Example
Labeled $d(v)/f(v)$

1/8 → 2/7 ← 9/10 → 12/13

4/5 ← 3/6 ← 11/16 → 14/15
Structure

Parenthesization
If we represent the discovery of vertex \( u \) with a left parenthesis “(\( u \)” and represent its finishing by a right parenthesis “\( u \)”), then the history of discoveries and finishings makes a well-formed expression in the sense that the parentheses are properly nested.

Parenthesis theorem In any depth-first search of a (directed or undirected) graph \( G = (V, E) \), for any two vertices \( u \) and \( v \), exactly one of the following three conditions holds:

- the intervals \([d[u], f[u]]\) and \([d[v], f[v]]\) are entirely disjoint, and neither \( u \) nor \( v \) is a descendant of the other in the depth-first forest,
- the interval \([d[u], f[u]]\) is contained entirely within the interval \([d[v], f[v]]\), and \( u \) is a descendant of \( v \) in a depth-first tree, or
- the interval \([d[v], f[v]]\) is contained entirely within the interval \([d[u], f[u]]\), and \( v \) is a descendant of \( u \) in a depth-first tree.

Nesting of descendants’ intervals
Vertex \( v \) is a proper descendant of vertex \( u \) in the depth-first forest for a (directed or undirected) graph \( G \) if and only if \( d[u] < d[v] < f[v] < f[u] \).
More Structure

White-path theorem

In a depth-first forest of a (directed or undirected) graph $G = (V, E)$, vertex $v$ is a descendant of vertex $u$ if and only if at the time $d[u]$ that the search discovers $u$, vertex $v$ can be reached from $u$ along a path consisting entirely of white vertices.

Edge classification

1. **Tree edges** are edges in the depth-first forest $G_\pi$. Edge $(u, v)$ is a tree edge if $v$ was first discovered by exploring edge $(u, v)$.

2. **Back edges** are those edges $(u, v)$ connecting a vertex $u$ to an ancestor $v$ in a depth-first tree. Self-loops, which may occur in directed graphs, are considered to be back edges.

3. **Forward edges** are those nontree edges $(u, v)$ connecting a vertex $u$ to a descendant $v$ in a depth-first tree.

4. **Cross edges** are all other edges. They can go between vertices in the same depth-first tree, as long as one vertex is not an ancestor of the other, or they can go between vertices in different depth-first trees.