Date: 05/05/21

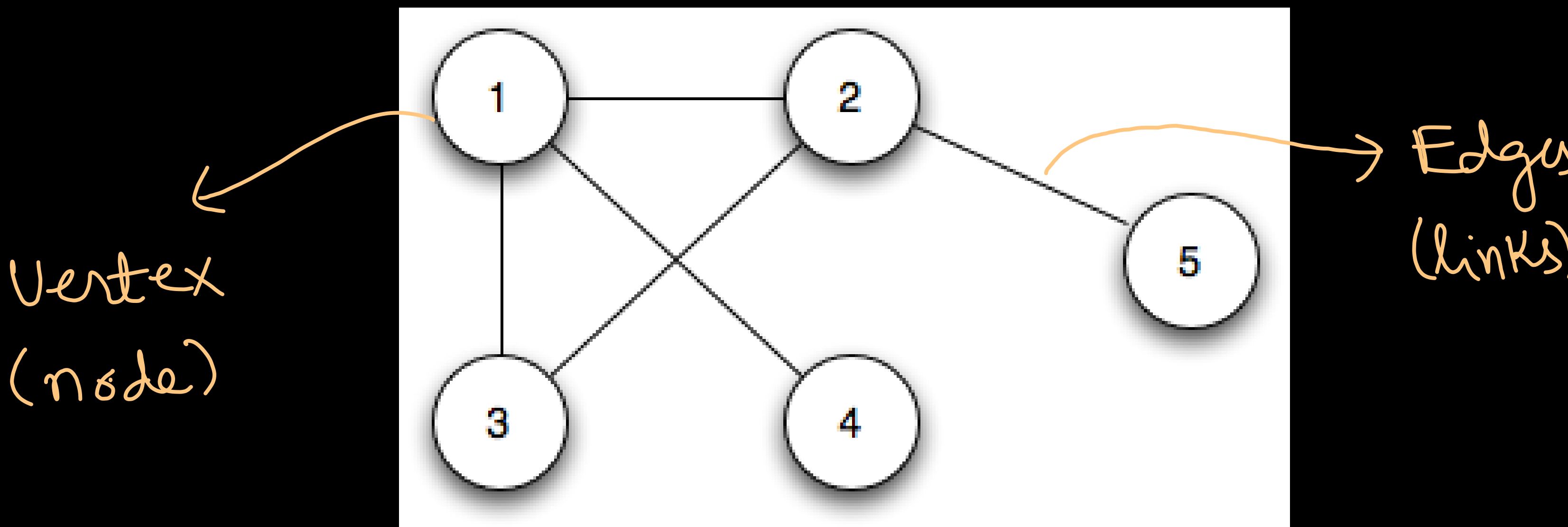
Black Board

Design and Analysis of Algorithms

Topics:

- Graph Algorithms I
  - Definitions
  - Representations
  - Depth First Traversal

What is a Graph?

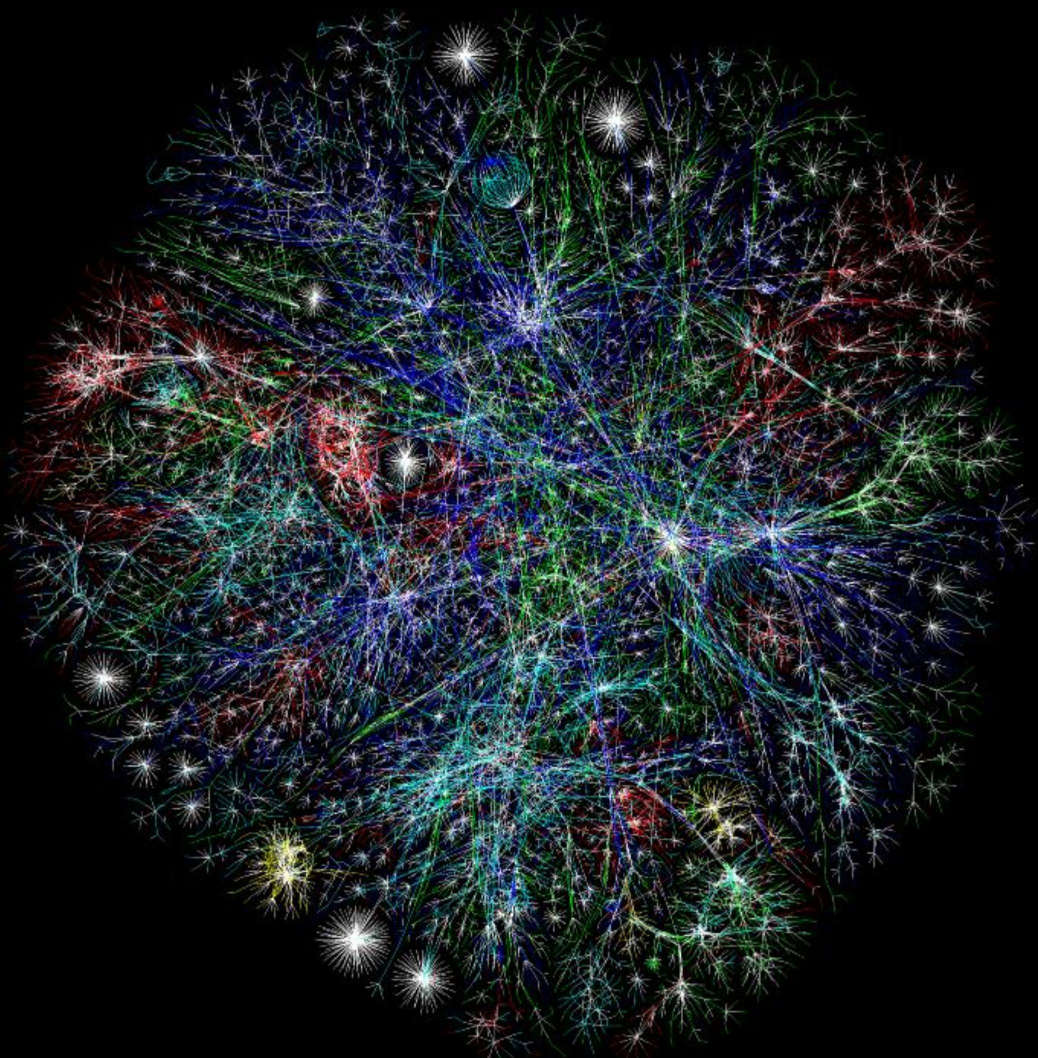


Commonly encountered Graphs in Computer Science

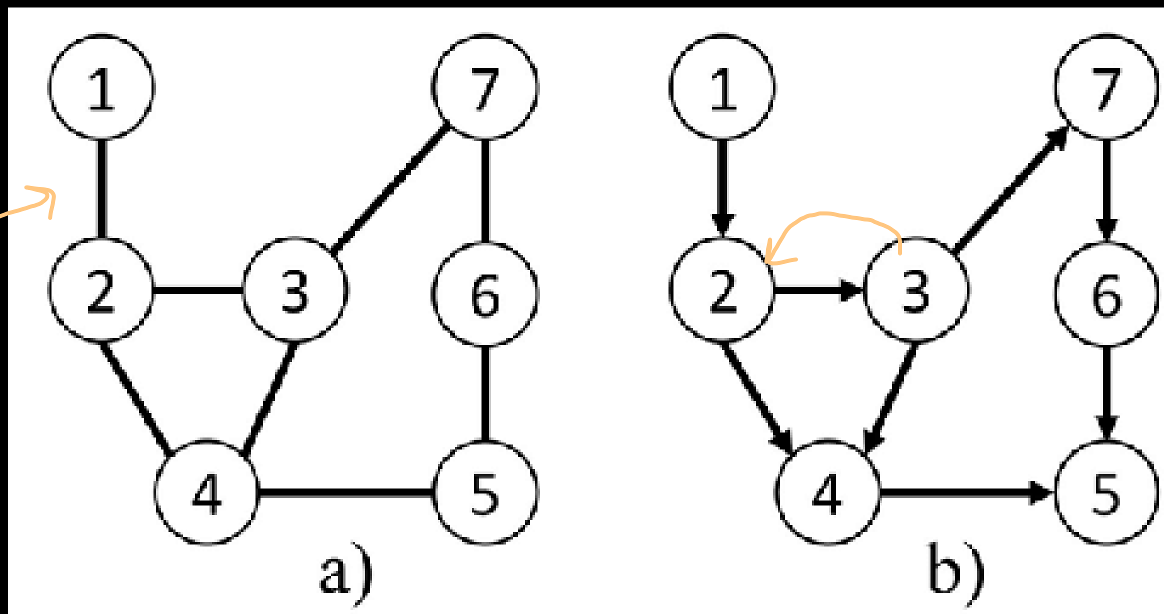
- The Internet
- Road Maps, e.g. Google Maps
- Social Media: Facebook, Twitter etc.
- Computer Networks: LANs, WANs, Overlay Networks, IOT etc.
- Less obviously: protein interaction networks in bioinformatics, semantic webs in AI, Image Maps in DIP etc.

What is this graph?

Internet



Graph Mathematical Notation
Directed and Undirected



$G = (V, E)$ $\quad V = \{1, 2, 3, 4, 5, 6, 7\}$

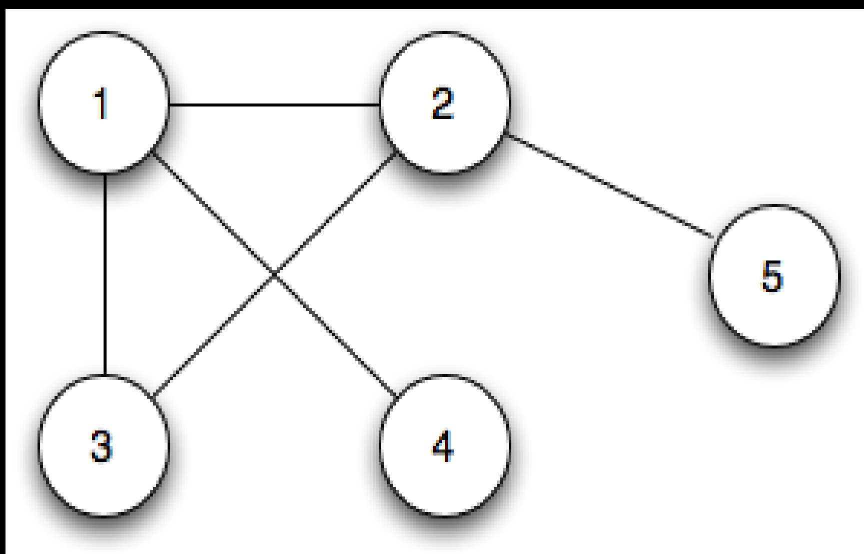
$E = \{\{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{3,7\}, \{6,7\}, \{5,6\}\}$

$G = (V, E)$ $\quad E = \{(1,2), (2,3), (3,2), \ldots\}$

Graph Representations

How to store graphs in programs?

- Adjacency Matrix



Sparse Matrix

$$
\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

Reverse in-place.

Space: $|V|^2$

Sparse Graph produces a sparse matrix.

$\longrightarrow G=(V,E)$

<u>Pro</u>: $O(1)$ edge lookup

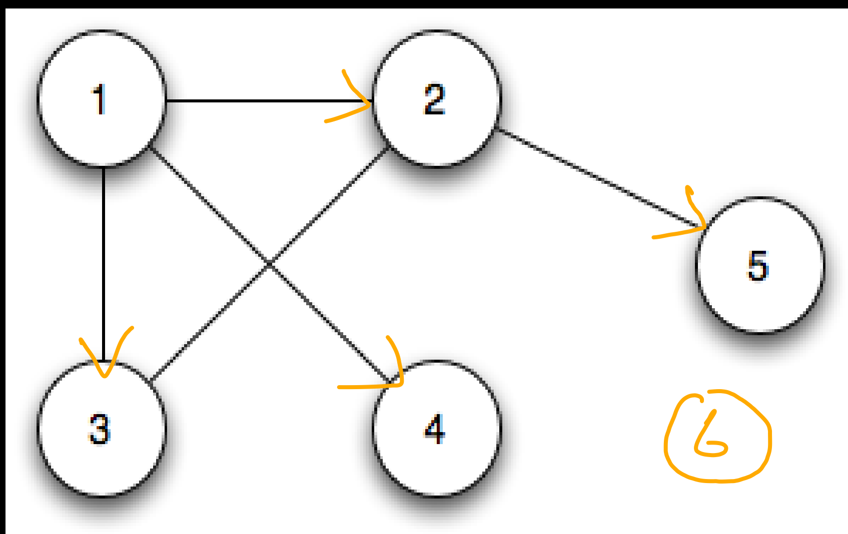
<u>Cons</u>: wasteful in terms of space.

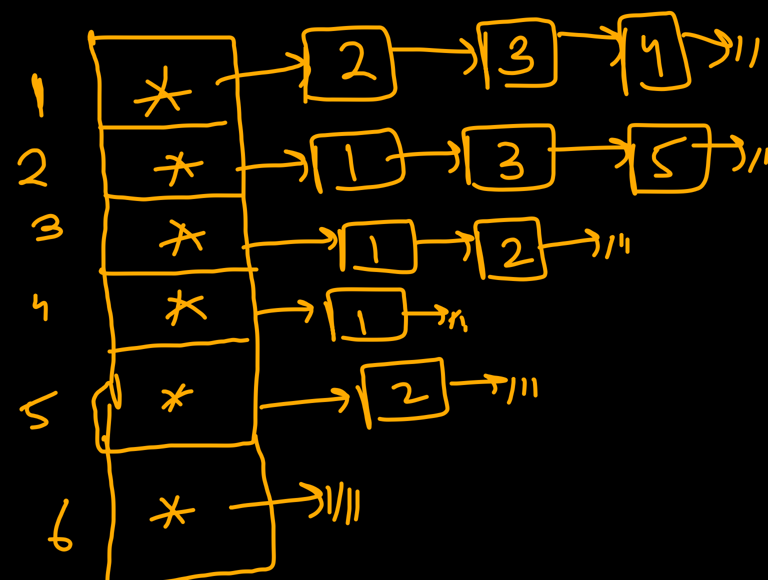
$G=(V,|E|)$ $O(|V|+|E|) \sim O(n)$, linear.

$O(|V||E|)$

- Adjacency List $O(|V|\lg|V|)$

Try reversing this graph In place. (possible?)





ReverseGraph$(G=(V,E))$

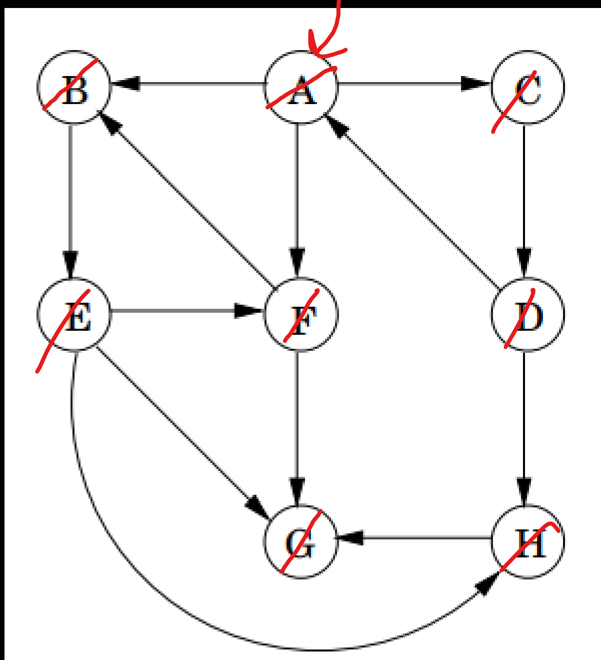
↳ $O(|V|+|E|)$

pros: space efficient

cons: No $O(1)$ edge access.

Traversing through a graph: the **explore** method

explore tree



Par[C] = A

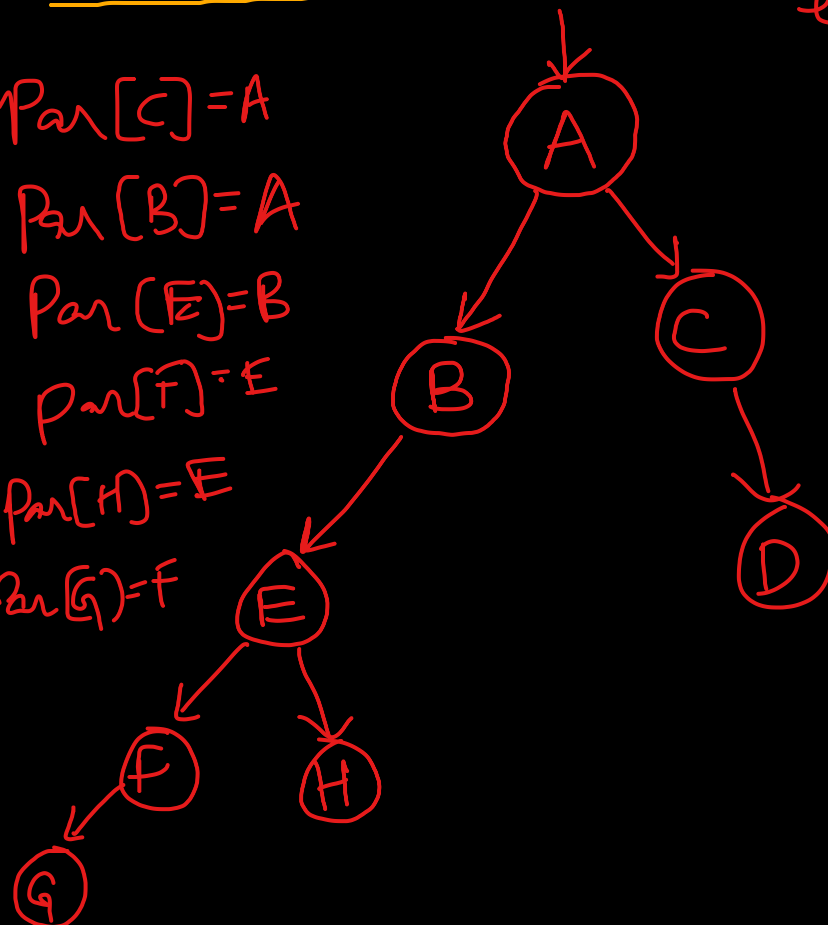
Par[B] = A

Par[E] = B

Par[F] = E

Par[H] = E

Par[G] = F



Par[D] = C

explore code

explore(G = (V, E), x)

{

visited[x] = true

Preprocessing( )

For each $(x, y) \in E$

IF(!visited[y]) Par[y] = x

explore(G, y)

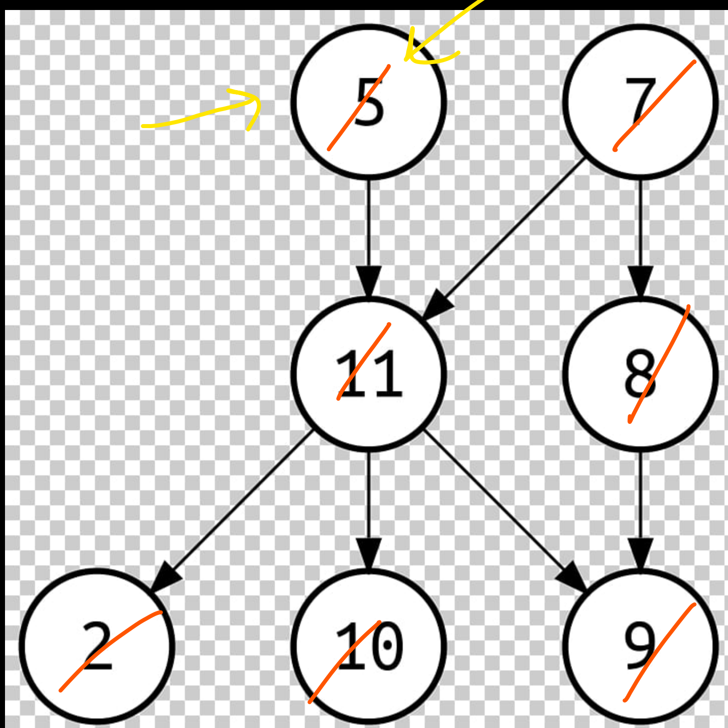
}

Postprocessing( )

}

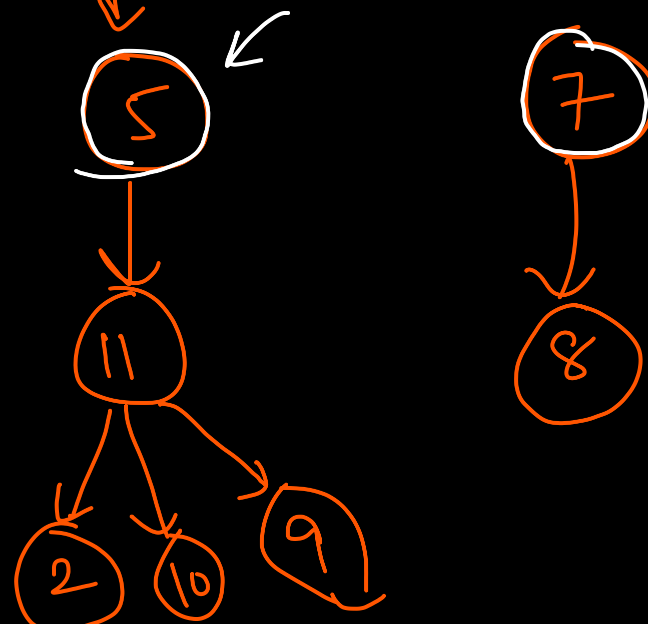
dfs using **explore**

We need multiple explores in general to visit every node once:



explore tree

explore .rec



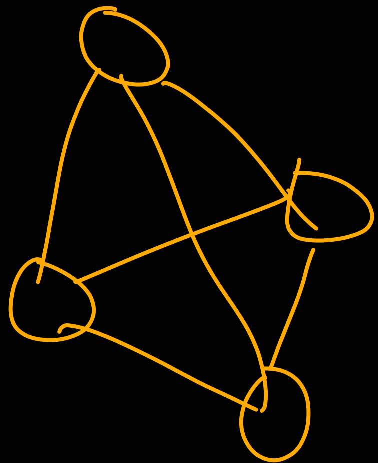
DFS Forest

dfs code

```
dfs( G=(V,E) )
{
    For each x ∈ V
        visited[x] = false

    For each x ∈ V
        IF (! visited[x])
            explore(G, x)
}
```

$|V|$ & $|E|$ relationship



Complete Graph

$$|E| \le \binom{|V|}{2}$$

Dense $\longrightarrow$

$\longleftarrow$ Sparse

$|E| = O(|V|^2)$

worst case

(Dense)

$$\boxed{O(|V| + |E|)}$$

Accurate for both Dense & Sparse Graph.