\( V \) - vertices, any finite nonempty set

\( E \) - edges = pairs of vertices

undirected: \( \exists u,v \notin E \setminus u \sim v \)

directed: \( (u,v) = u \rightarrow v \)

intersection graphs
\[ F_n = F_{n-1} + F_{n-2} \]

Edit distance $\Rightarrow$

Directed acyclic graphs
Data structures

v. color
color[v]
mark
mark[v]

**DEFAULT:**

adjacency list
check if u,v ∈ E  \( O(\min(\deg(u), \deg(v))) \)
list all neighbors of \( u \) in \( O(\deg(v)) \)
\( O(V+|E|) \) space

**Adjacency matrix:**
u,v ∈ E?  \( O(1) \) time
list all neighbors of \( v \)  \( O(|V|) \) time
\( O(V^2) \) space
**Depth-First Search**

**RecursiveDFS(v):**
- if v is unmarked
  - mark v
  - for each edge vw
  - RecursiveDFS(w)

**IterativeDFS(s):**
- Push(s)
- while the stack is not empty
  - v ← Pop
  - if v is unmarked
    - mark v
    - for each edge vw
    - Push(w)

**WhateverFirstSearch(s):**
- put s into the bag \( \leq 1 \)
- while the bag is not empty
  - take v from the bag
    - if v is unmarked
      - mark v
      - for each edge vw
      - put w into the bag

\[ O(V + E) \]

**WhateverFirstSearch(s):**
- put \((\emptyset, s)\) in bag
- while the bag is not empty
  - take \((p, v)\) from the bag
    - if v is unmarked
      - mark v
      - parent(v) \(\leftarrow p\)
      - for each edge vw
        - put \((v, w)\) into the bag

\((\ast)\)

\((\dagger)\)

\((\ast\ast)\)
WFSALL\((G)\) :
for all vertices \(v\)
unmark \(v\)
for all vertices \(v\)
if \(v\) is unmarked
  \[\text{WHATEVERFIRSTSEARCH}(v)\]

COUNTCOMPONENTS\((G)\) :
\[\text{count} \leftarrow 0\]
for all vertices \(v\)
  unmark \(v\)
for all vertices \(v\)
if \(v\) is unmarked
  \[\text{count} \leftarrow \text{count} + 1\]
  \[\text{WHATEVERFIRSTSEARCH}(v)\]
\[\text{return count}\]

\[O(V+E)\] time

COUNTANDLABEL\((G)\) :
\[\text{count} \leftarrow 0\]
for all vertices \(v\)
  unmark \(v\)
for all vertices \(v\)
  if \(v\) is unmarked
    \[\text{count} \leftarrow \text{count} + 1\]
    \[\text{LABELONE}(v, \text{count})\]
\[\text{return count}\]

LABELONE\((v, \text{count})\) :
while the bag is not empty
take \(v\) from the bag
if \(v\) is unmarked
  mark \(v\)
  \[\text{comp}(v) \leftarrow \text{count}\]
  for each edge \(vw\)
    put \(w\) into the bag

LABELONE\((v, \text{count})\) :