BFS DFS

Graph Traversal Algorithms

- BFS
- DFS

white : unexplored
gray : being explored
black : finished exploring

⟹ **BFS**

BFS(G, s) ← source

<u>initialization</u> : for u in $V \setminus \{s\}$:
    color[u] = white
    (parent) π[u] = nil
    d[u] = ∞
$\Big\}$ O(n)

Enqueue (Q, s) ; color[s] = gray ; π[s] = nil, d[s] = 0
while Q ≠ ϕ
    u = Dequeue(Q)
    for v in Adj[u] : ← O(degree of u)
        if color[v] == white
            color[v] = gray
            d[v] = d[u] + 1
            π[v] = u
            $\Big\}$ O(1)
            Enqueue (Q, v)
    color[u] = black

∴ O(sum of degree)
= O(2m)
= O(m)

∴ $\boxed{O(m+n)}$

> Property: d[v] holds the value of the shortest path from s to v

<u>lemma</u>: $s \in V$ $(u,v) \in E$ $\delta(s,v) \le \delta(s,u) + 1$ [Directed / Undirected]

[$\delta(s,v)$: Shortest path from s to v]

if u is reachable from s, so is v via path from s to u then u to v
if u is not, v is also not reachable, d(s,u) = d(s,v) = ∞, equality holds

lemma: d[v] $\ge \delta(s,v)$

induction of # Enqueue Operations

indn hypothesis: $d[v] \geq \delta(s,v) \quad \forall v \in V$

Base Case: $d[s] = \delta(s,s) = 0$

$d[v] = \infty \geq \delta(s,v) \quad \forall v \in V$] at first Enque Operation

Inductive Step: Consider v is discovered during the search from u.

$d[u] \geq \delta[s,u]$ } u is Enqueued before induction hypothesis

$d[v] = d[u] + 1$ (algorithm)

$\geq \delta(s,u) + 1$ (previous lemma)

$\geq \delta(s,v)$

v → Enqueued, becomes gray, so it is never enqued again, so d[v] is not changed again.

∴ Induction hypothesis maintained.

Lemma: Suppose Q contains $\{v_1, v_2, \ldots, v_r\}$ head tail

$d[v_i] \leq d[v_{i+1}]$

$d[v_r] \leq d[v_1] + 1$

induction of # queue operations

Base Case: initially $Q = \{s\}$, true

indn step: (1) when Deque (Q, v_1) ; v_2 becomes head

$d[v_1] \leq d[v_2]$ (b. ind^n hypothesis)

$\leq d[v_2]$ (by the hypothesis)

$d[v_r] \leq d[v_1]+1 \leq d[v_2]+1$

$\therefore \underbrace{d[v_r] \leq d[v_2]+1}_{\text{new head}}$

(2) Enque (Q, v_{r+1})

if previously $Q = \phi$, trivially true

else previously Q was not empty before.

$\exists\, v$, which was Dequeued and $Adj[v]$ is being explored.
Just before v was removed $v = v_1$ (v is head)

$$\left.\begin{aligned} d[v] &\leq d[v_2] \\ d[v_r] &\leq d[v]+1 \end{aligned}\right\} \text{ indn hypothesis}$$

After removing v; v_2 is head

$v_2 = v_1$ (v_2 is head)

$$\left\{\begin{aligned} &\underbrace{d[v_{r+1}] = d[v]+1} \\ &\text{from algorithm} \\ &v \text{ is dequeued} \\ &\text{and } v_{r+1} \in Adj[v] \end{aligned}\right.$$

$d[v] \leq d[v_1]$

$\underbrace{d[V]}_{\text{newly enqueud}} = d[v_{r+1}] = d[v]+1 \leq d[v_1]+1$

$d[v_r] \leq d[v]+1 = d[V] = d[v_{r+1}]$

$\therefore$ inequalities unaffected

Corollory: v_i enqueued before v_j $\quad d[v_i] \leq d[v_j]$

from previous lemma

$\langle v_1, v_2, \dots, v_r \rangle$ $\quad d[v_1] \leq d[v_2] \dots$

$\langle v_1, v_2, \ldots, v_r \rangle \quad d[v_i] \leq d[v_{i+1}] \leq d[v_{i+2}] \leq \ldots \leq d[v_j]$

v_i enqueued before v_j

+

property: $d[v]$ is changed only once.

<u>Theorem</u>: Correctness of BFS

Assume contradiction, $\exists\, v \in V$ such that $d[v] \neq \delta[s, v]$

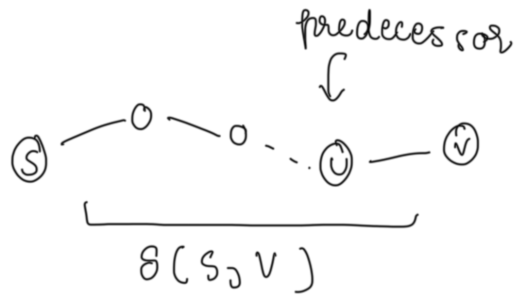
Let v be the vertex with minimum $\delta(s, v)$ among "bad" vertices

$d[v] \geq \delta[s, v]$ (lemma)

so $d[v] > \delta[s, v]$

$d[u] = \delta[s, u]$

(u is a "good" vertex)



$d[v] > \delta[s, v] = \delta[s, u] + 1 \quad \rightarrow \quad \star$

now;

consider when u is dequeued

Case 1: v is white

$d[v] = d[u] + 1$, contradicts $\star$

Case 2: v is gray.

then $\exists\, w \quad \pi[v] = w$

$d[v] = d[w] + 1$

$d[w] \leq d[u]$ [Corollary, w enqueued before u]

$d[v] = d[w] + 1 \leq d[u] + 1$,

contradicts $\star$

Case 3: v is black.

v is dequeued before u

$d[v] \leq d[u] + 1$ (by lemma)

contradicts $\star$

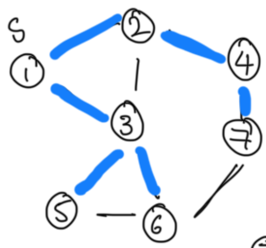
$\therefore d[v] > \delta[s,v]$ is a contradiction

$\therefore d[v] = \delta[s,v]$

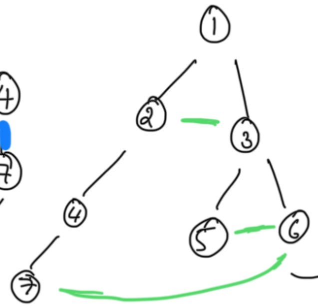
BFS tree

for directed

BFS tree:



Tree Edges

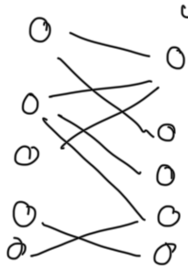


cross edges

within same level: odd length cycle detected, not bipartite.

→ if $\exists (u,v) \in E$
u and v are at the same level or one level apart

Bipartiteness: bipartite $\Leftrightarrow$ no odd length cycles



isBipartite (G)

```
color[V] = -1
for i = 0 to n
    if (color[V_i] ≠ -1) continue
    Queue <int> Q
    Enqueue (V_i)   color[V_i] = 0
    while (Q)
        u = Deq (Q)
        for v ∈ Adj[u]
            if (color [V] ≠ -1)
                color[V] = flip(color[U])
            else if color[V] = = color[U]
                return false
return true
```

$O(m+n)$

$\Rightarrow$ DFS

DFS(G)

for $u \in V$
  color[u] = white
  π[u] = nil

time = 0
for $u \in V$
  if color[u] == white
    DFS_Visit(G, u)

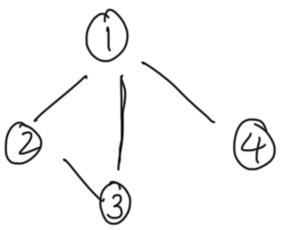
called only once per vertex $v \in V$
$\therefore O(\Sigma|Adj(v)|) = O(m)$

DFS_visit(G, u)

time ++
d[u] = time (u discovered) $\rightarrow$ discovery time
color[u] = gray
for v in Adj[u]
  if color[v] == white
    π[v] = u
    DFS_visit(G, v)

Time ++
f[u] = time (u finished) $\longrightarrow$ finish time
color[u] = black

eg.



d[1] = 1
π[2] = 1 d[2] = 2
π[3] = 2 d[3] = 3 f[3] = 4
f[2] = 5
π[4] = 1 d[4] = 6 f[4] = 7

> Properties
(1) Predecessor graph forms a forest
$E_\pi := \{(\pi[u], u) : u \in V \quad \pi[u] \neq NIL\}$

$G_\pi := (V, E_\pi)$

each tree, (u,v) when (u,v) explored u is grey, v is white

$u = \pi[v]$ iff DFS_visit $[v]$ called during a search of u's adjacency list

(2) Paranthesis Theorem.

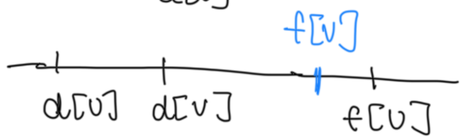
(1) $(d[u], f[u])$ $(d[v], f[v])$ disjoint

(2) $(d[u], f[u])$ contained in $(d[v], f[v])$

(3) $(d[v], f[v])$ " " $(d[u], f[u])$

Proof wlog $d[u] < d[v]$

Case 1 $d[v] < f[u]$



v is a decendent of u

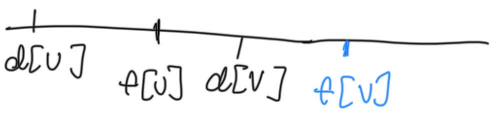
v is white

u is gray

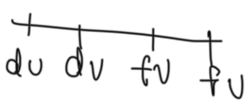
u becomes black

so after v's outgoing edges are visited & v is finished u can finish

Case 2



Corollary v is a proper decendent of u $\Longleftrightarrow$



$\Rightarrow$ White Path Theorem

v descendent of u $\Longleftrightarrow$ at $d[u]$ $\exists\, u \rightsquigarrow v$

DF ~~first~~ 1.

white vertices

$\Longrightarrow$ $v = u$ trivial

Suppose v is a proper descendant of u

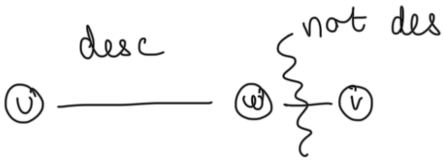
$d[u] < d[v]$

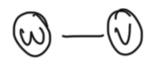
at $d[u]$ v is white

all descendents in the unique simple path in DF Forest are white.

$\Longleftarrow$ $\exists$ white path

Let v be closest vertex on this path to not be descendent



since 

v discovered before w finishes

$f[w] < f[u]$

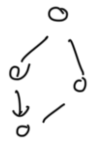
$d[u] < d[v] < f[w] < f[u]$

$\therefore$ v contained in $w \to$ descendent

$\Rightarrow$ Classification of Edges

directed graph acyclic $\Longleftrightarrow$ no back edges

1. Tree Edges
2. Back edges
3. Forward Edges

4. Cross Edges 

Undirected graph:

Every edges is tree or

back.

(u, v) first explored

white (tree edge)
gray (back edges)
black (forward / cross)

Applications BFS, DFS

$\Longrightarrow$ **Topological Sort**

$\boxed{O(m+n)}$

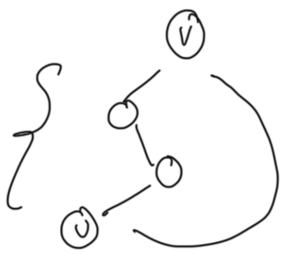
Topological Sort (G)

1. DFS (G)
2. as each vertex is finished, insert to front of linked list
3. Return linked list

__Lemma__ G (directed) acyclic $\Longleftrightarrow$ DFS yields no back edges

__Proof__ $\Longrightarrow$ If back edge (u, v)

if DF tree v is ancestor of u

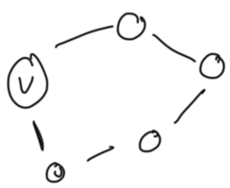


back edges cycle

back edge → cycle
∴ acyclic → no back edge

$\Longleftarrow$ Suppose ∃ cycle C

first vertex to be discovered →



at d[v] ∃ white path from v to u
∴ u is a descendent of v.
∃ back edges (u, v)

cycle $\rightarrow$ back edge

$\sim$ back edge $\rightarrow$ $\sim$ cycle

Proof of Correctness:

DAG $G(V, E)$

$u, v \in V$ if $u \rightarrow v$ $\rightarrow$ $\underbrace{f[v] < f[u]}$ f[v] f[u]

u comes earlier in topological sort

consider any edge $[u\ v]$ explored.

Case 1) v gray. not possible, otherwise it would be a back edge implying cycle.

Case 2) v white

v is a descendent of u

d[u] d[v] f[v] f[u]

$f[v] < f[u]$

case 3) v black

v finished earlier

$f[v] < f[u]$

$\Longrightarrow$ SCC

SCC:

(maximum subset of connected vertices)

DFS(G)

G^T

DFS(G^T) in order of finish times

Output each DFS tree as a SCC

Proof

$SCCs(G) = \{C_1, \ldots, C_k\}$

$d(C) = \min\{d(u) \mid u \in C\}$

$f(C) = \max\{f(u) \mid u \in C\}$

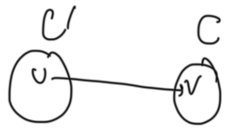
<u>Lemma</u> $C \neq C'$

$u, v \in C \quad u' \; v' \in C'$

G contains $u \rightsquigarrow u'$

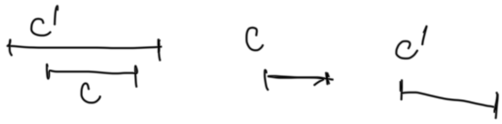
G cannot contain $v' \rightsquigarrow v$

proof: contradicts $C \neq C'$



<u>Lemma</u> $(u, v) \in E \quad u \in C' \quad v \in C$

$f(C') > f(C)$



<u>Proof</u> (if) $d(C') < d(C)$

let $d(C') = d(x)$

$\exists$ white path from x to all nodes in C

$\therefore \; f(x) = f(C') > f(C)$

(else) $d(C') > d(C) = d(y)$

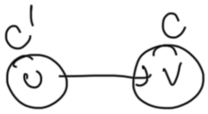
at $d[y]$

C' is unvisited and white

DFS from y cannot reach C'

$\therefore \; f[C] < f[C']$

<u>Corollary</u> $f[C] > f[C']$ no edge $(v, u) \in E^T$ such that $v \in C$ $u \in C^T$



$\longrightarrow \; f[C] < f[C']$

$f[C] > f[C'] \longrightarrow$ no edge $uv \in E$

no edge $vu \in E^T$

<u>Proof of Correctness</u>:

Let DFS (G^T) ~ $T_1, T_2, \ldots, T_k$
trees

<u>Base Case</u> $k=0$
trivially true

<u>Inductive Step:</u> $k \to k+1$

Let u be root of $(k+1)^{th}$ tree

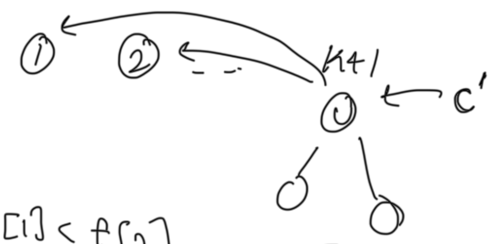
$v \in C$

(1) $f(C) > f(C')$ $\forall$ unvisited C'

(2) White path theorem ; nodes in C become descendents of v

(3) Corollary

no outgoing edges to unvisited components



DFS tree rooted at v
exactly vertices in C.

$f[1] < f[2] < f[C]$