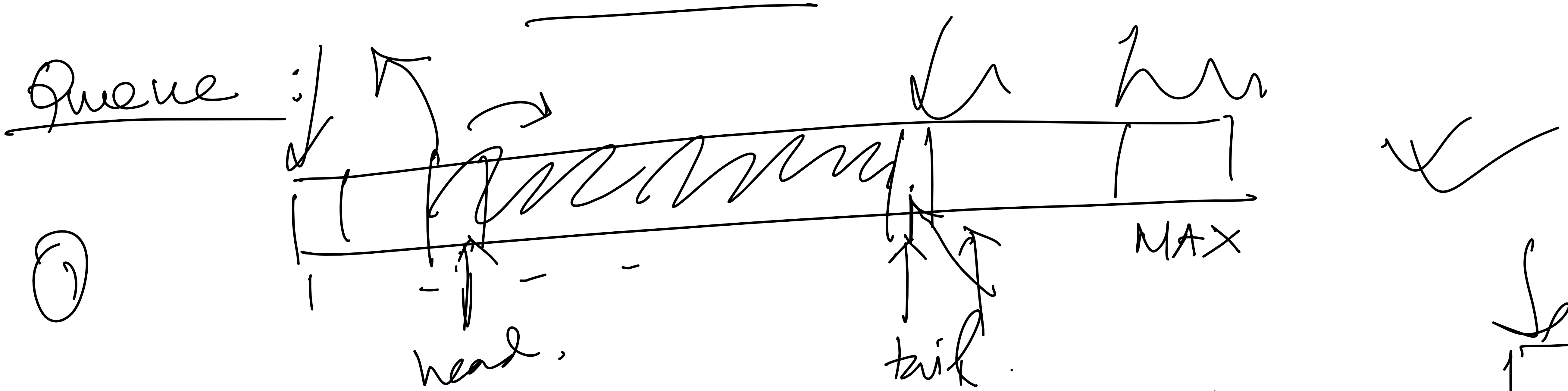
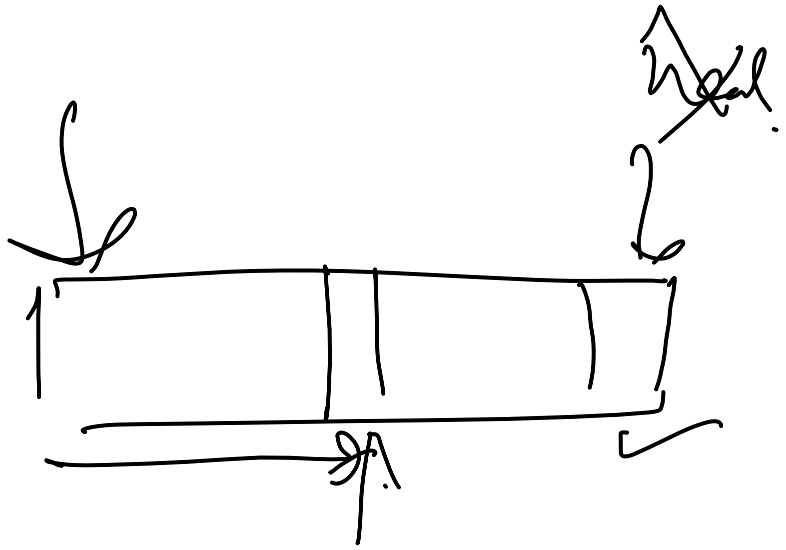
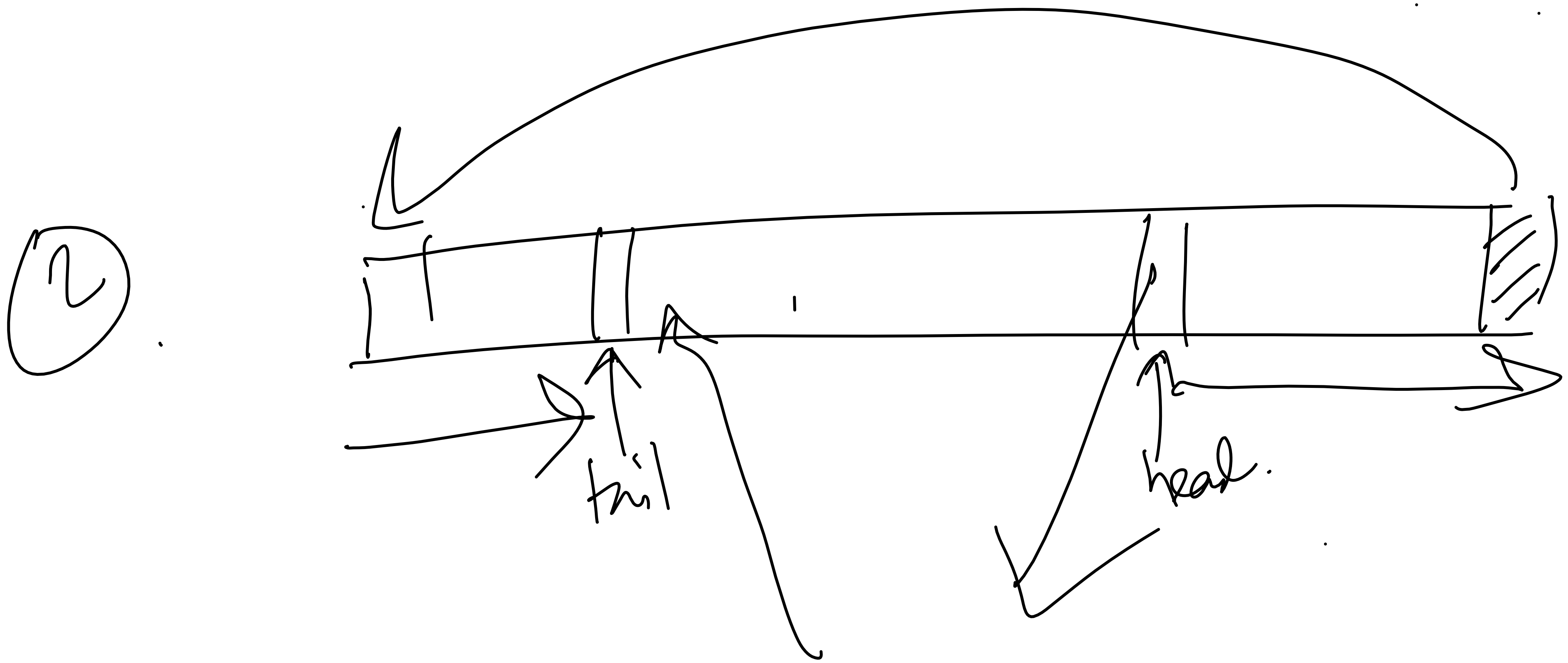
21.08.2024

: Queue :







initially, Q.head = Q.tail = 0.

```
Procedure IsEmpty (Q)
    if (Q.head = Q.tail) return "yes"
    else    return "no"
    end if
```

```
Procedure IsFull (Q)
    ~~if (Q.tail > Q.head)~~
    if IsEmpty (Q) = yes    return "no"
    else if (Q.tail > Q.head)
        if (Q.head = 1 & Q.tail = max)  return yes
        else return no.
```

```
    else if (Q.tail < Q.head)
        if (Q.head = Q.tail + 1) return "yes"
        else return "no"
```

```
Procedure Enqueue (Q, a).
    if (IsFull(Q) = yes) return "error" endif
    Q.[Q.tail + 1] ← a.
    if (Q.tail = MAX) Q.tail ← 1
    else Q.tail ← Q.tail + 1
    endif
```

Procedure Dequeue (Q).

if (IsEmpty (Q) = yes) return error.

temp ← Q[Q.head].

if (Q.head = MAX) Q.head = 1

else Q.head = Q.head + 1

return temp

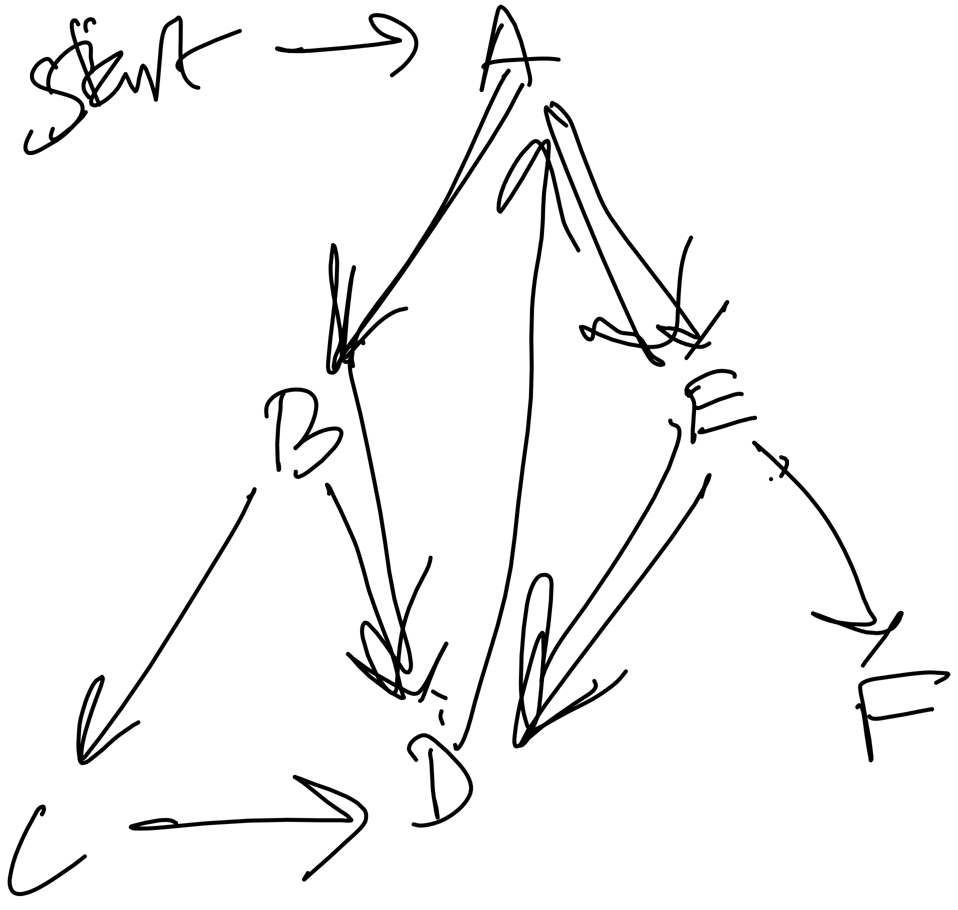
FIFO = First in first out : Queue

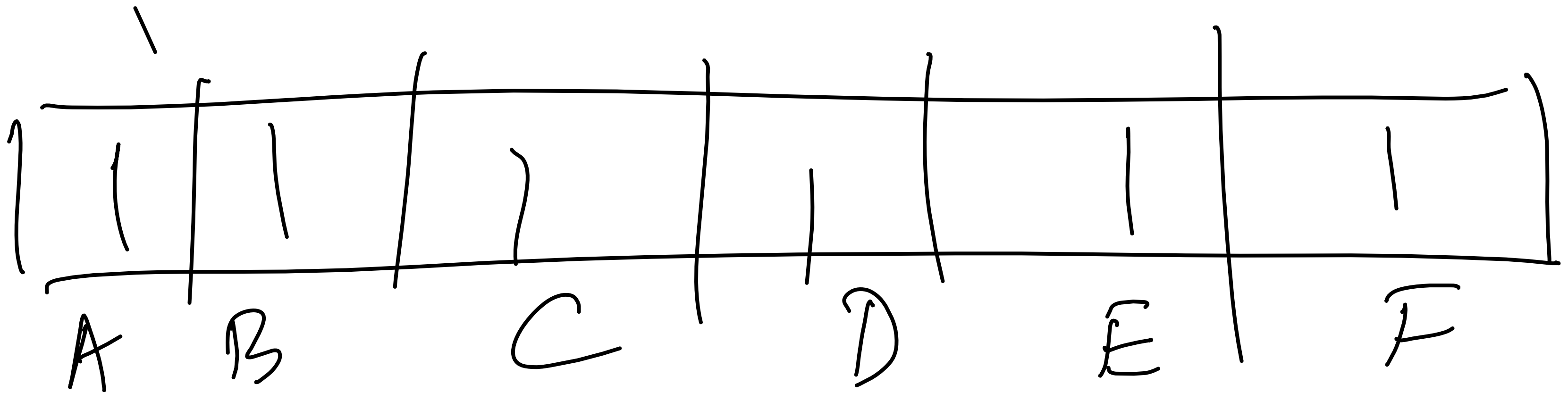
LIFO = Last in fm out. : Stack.

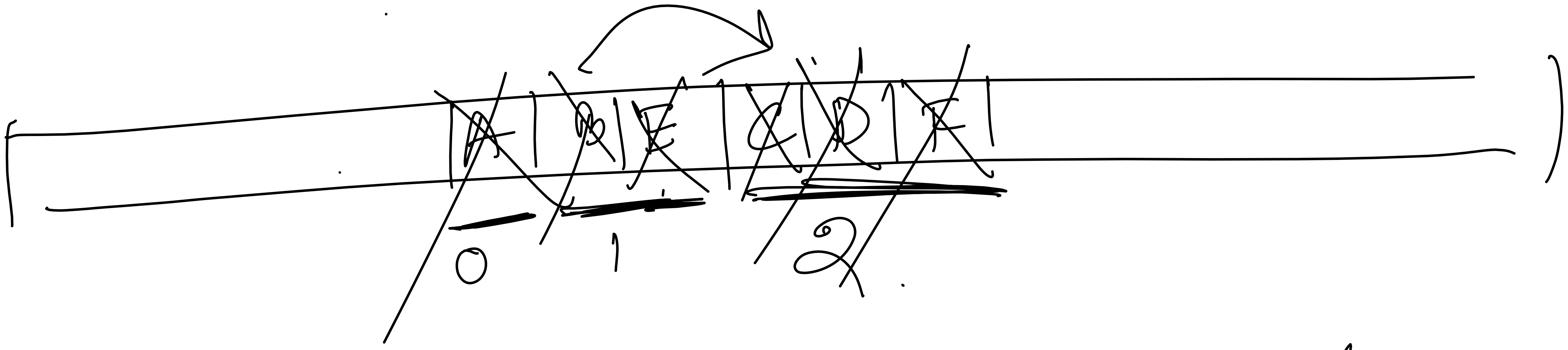
$V_i \longrightarrow V_j$

(1) Is V_j readable from V_i?

✓ - If yes, what is shortest hop path.







Shortest path length is unique but shortest path itself need not be.

Procedure BFS (G, v)

$\downarrow$ Start vertex

// $G = (V, E)$

~~A ← all~~
~~for each~~

$\Pi \leftarrow$ empty array of size $|V|$. $\Theta(n)$

$A \leftarrow$ all 0 array of size $|V|$. $\Theta(n)$

$Q \leftarrow$ empty Q, with MAX $= |V|$ $\Theta(n)$

Enqueue (Q, v). — $\Theta(1)$.

while (IsEmpty (Q) = NO). runs $\leq n$ time.

~~can iterat~~

at most 1 iteration per vertex

$u \leftarrow$ Dequeue (Q). $\Theta(1)$

for each $(u, w) \in E$

n, m

$\boxed{\text{if } (A(w) = 0)}$

$A(w) = A(u) + 1$ ✓

$\pi(w) \leftarrow u$.

Enqueue (Q, w).

end for

end while.

<u>Correctness</u>. ~~Start~~ vertex is fixed to s

<u>Claim</u>: Let $A_{\leq d} = \{ w : \cancel{level}\, A(w) \leq d \}$

$B_{\leq d} = \{ v : \underbrace{dist(v)}_{\text{length of shortest path from } s \text{ to } v} \leq d \}$

Then, $A_{\leq d} = B_{\leq d}$

<u>Proof</u>: By Induction.

<u>B.C.</u> <u>$d = 1$</u> $A_{\leq d}$: s & and its children.

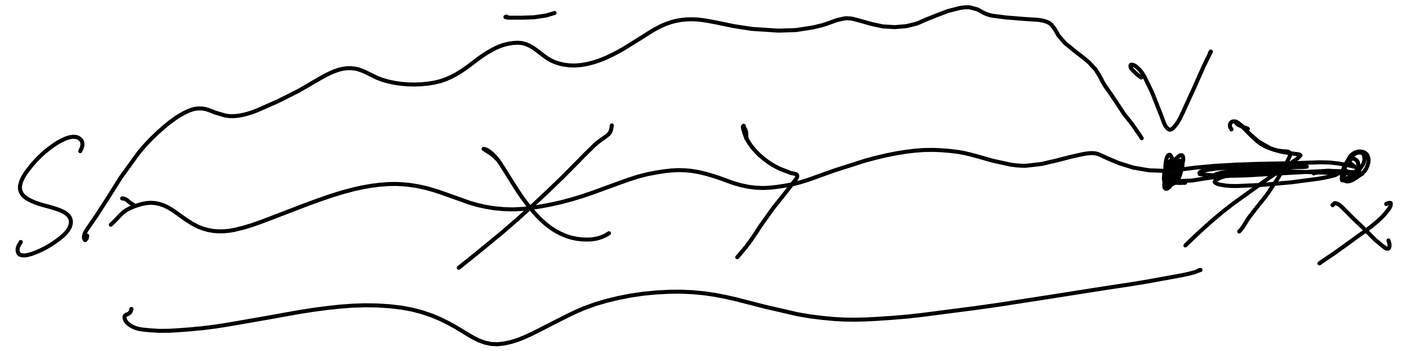
$= B_{\leq d}$

I.H.: suppose holds for $(d-1)$:

I.S.: $A_{\le d} \subseteq B_{\le d}$. [Trivial].

Now we'll show $\boxed{B_{\le d} \subseteq A_{\le d}}$

Let $x \in B_{\le d}$.

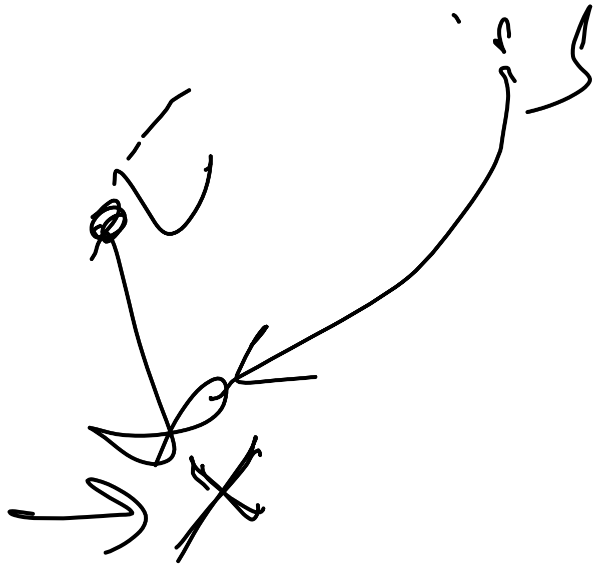


$\le d$.

Importantly: $s \rightsquigarrow v$ path also has to be shortest.

$$v \in B_{\le(d-1)} = A_{\le(d-1)}$$

This means $A[v] \le (d-1)$



$A[w] = ?$

$\le (d-1) + 1$

$\le d$.

$\Rightarrow x \in A_{\le d}$.

Therefore $\boxed{A_{\le d} = B_{\le d}}$.

$\square$

Corollary: $\boxed{A_{\le d} \setminus A_{\le d-1}} = B_{\le d} \setminus B_{\le (d-1)}$

Time Complexity

Count how many times (*) step runs.

$$\sum_{v} \underbrace{\mathrm{Out}_v}_{\text{Outdegree of } v}$$

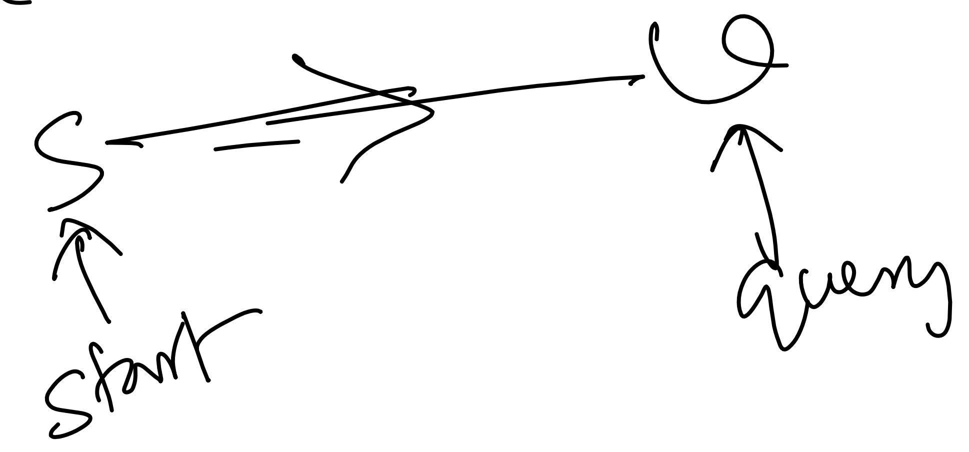
$= \#\text{edges} = m$. [exercise].

undirected graph $= 2m$.

Remark:

$\Theta(n + m)$ Same as DFS.

Recovery of path



Give me the shortest path from s to v.

Procedure - - - (s, v).

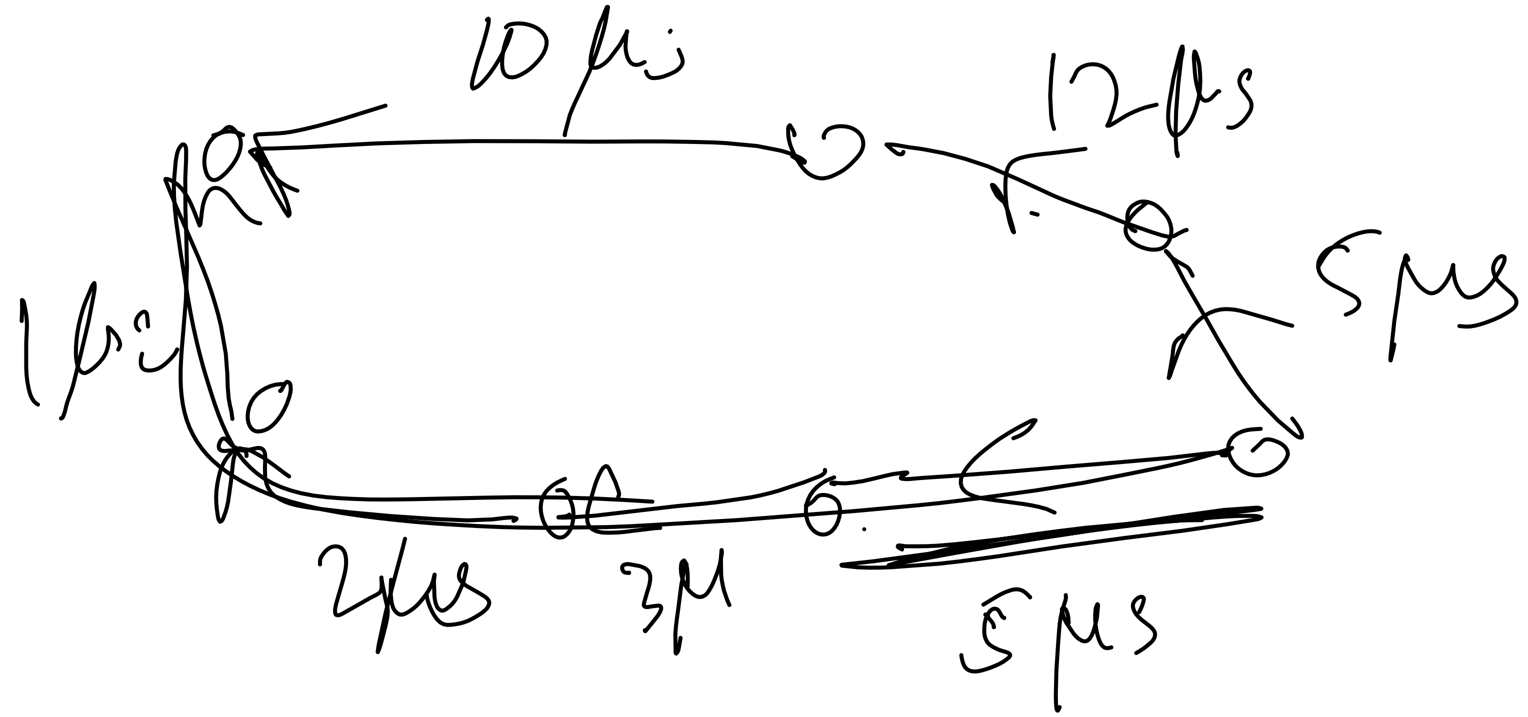
$u \leftarrow v$

While $(u \neq s)$.

  Print$(u, \pi(u))$

  $u \leftarrow \pi(u)$.

Single Source Shortest Paths.


10 μs
12 μs
5 μs
1 μs
2 μs
3 μ
5 μs