(5)

1) Graphs what?
Repre - Adj List
- Adj mat

2) DP

1) Recursion
2) optimization → memoize
3) space optimize

① Trav → DFS BFS

1) numbers of isle
2) matrix
2

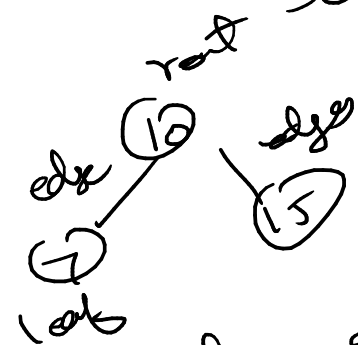
1) cycle — directed / undirected
2) Topological
3) course schedule

3) shortest path
Dijkstra
4) flod warshall

Graph

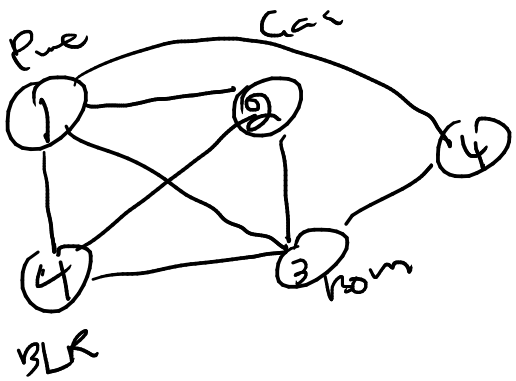
set of vertices & edges

Tree

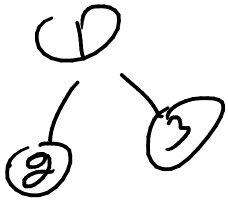


node edge

n → n-1



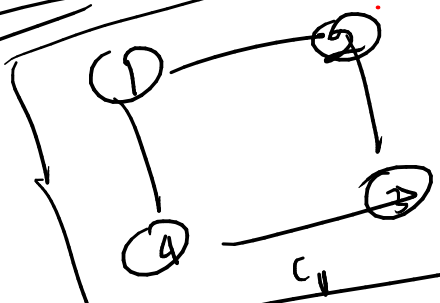
n(E) → ?
n(V) = 5

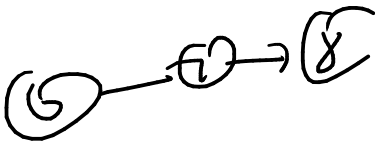


Tree → graph n / $n-1$

graphs → trees

concepts of graph not diff





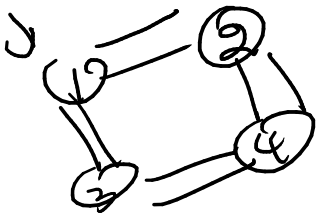
C2

C1

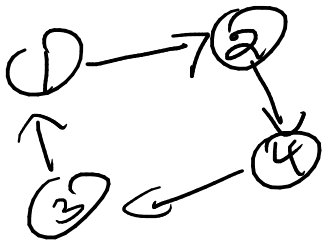
simple graph

handshake

$u \to 1$ $e \to 2v$ $u = 8$

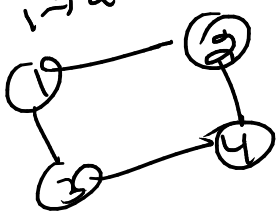


Directed



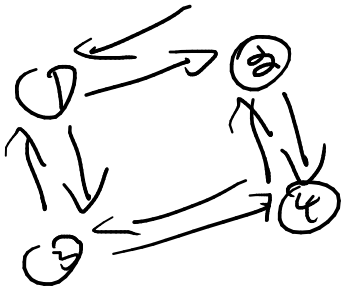
undirected graph

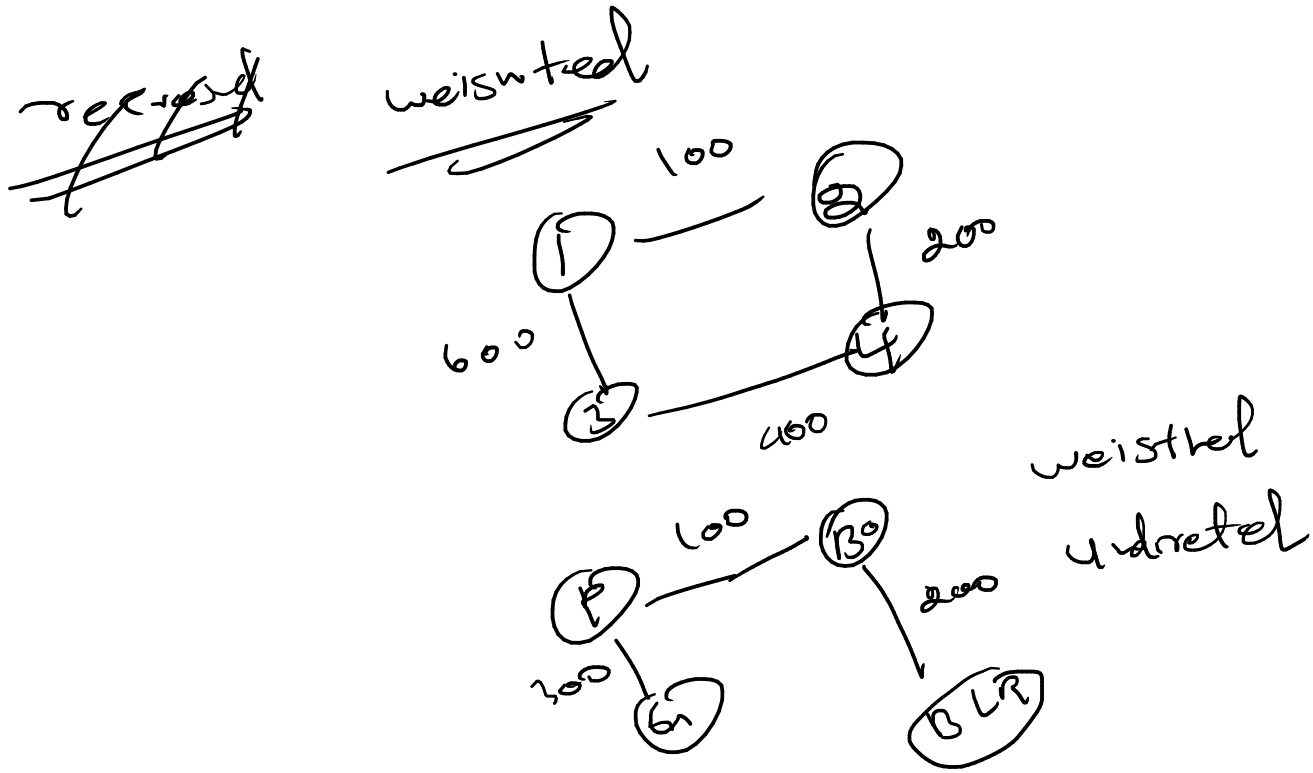
$1 \to 2$ $2 \to 1$

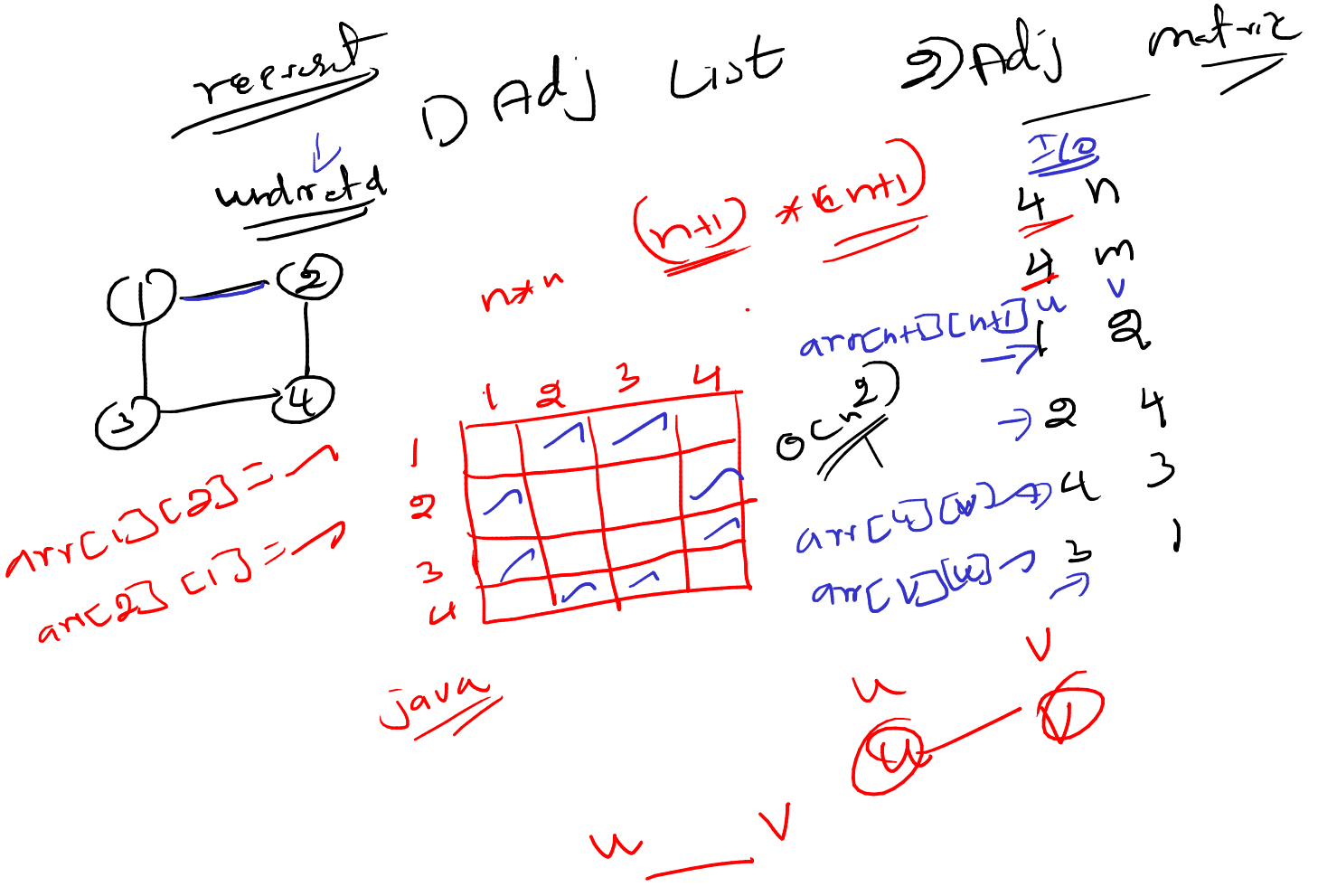


$1 \to 2$

$2 \to$ ✗




weighted
100
200
600
400
weighted
undirected
100
200
300

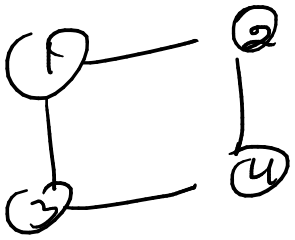

represent
Adj List
Adj matrix
undirected
(n+1) * (n+1)
n*n
I/O
4 n
4 m
u v
1 2
2 4
4 3
3 1
1 2 3 4
Java
u
v

1 2 3 4

1 → 2

2 → 1

adj



1 → 2 3

2 → 1 4

3 → 1

4 → 2 3

4

4

1 2

2 4

4 3

3 1

u → v

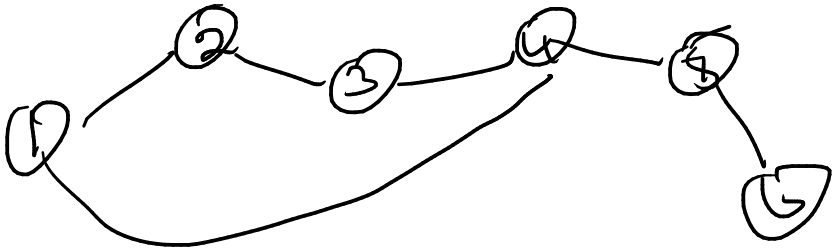
v → u

undirected

1 → n

n * 2m

h2



n2

n + 2m

O(n)

1 → 2

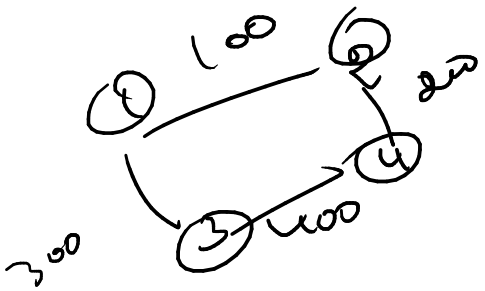
2 → 1 3

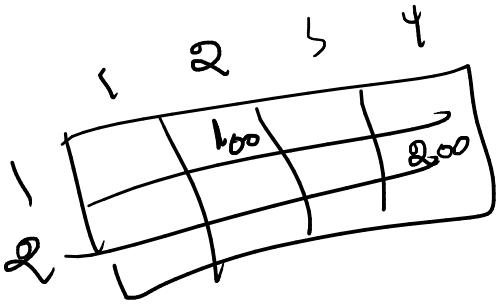
3 → 2 4

4 → 3 5

5 → 4 6

6 → 5





4

1 2 100

2 4 200

1 → {2, 100} {3, 300}

2 → {1, 100} {4, 200}

{4, 400} {1, 300}

3 4 400 → 3 → 200 → {3, 400}

1 3 300 → 4 → 2

Traversal

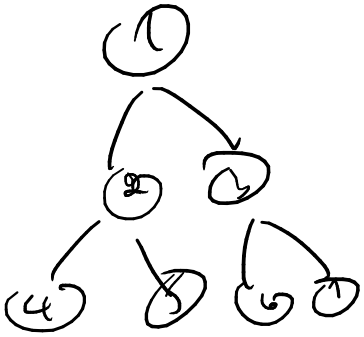
Level ord

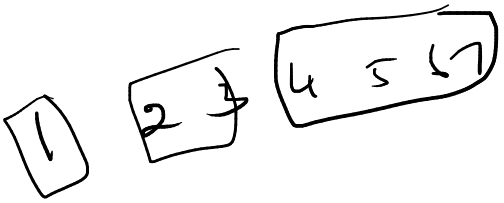
BFS (Breath First Seat)

DFS (Depth ———)

↓

Ino | Pre | Po





AS son → go inside

Adj

DFS

1 2 5

3 1

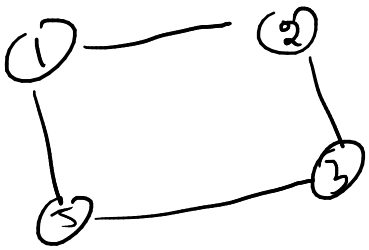
2

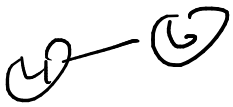
3 2 5

4 6

5 1 3

6 4

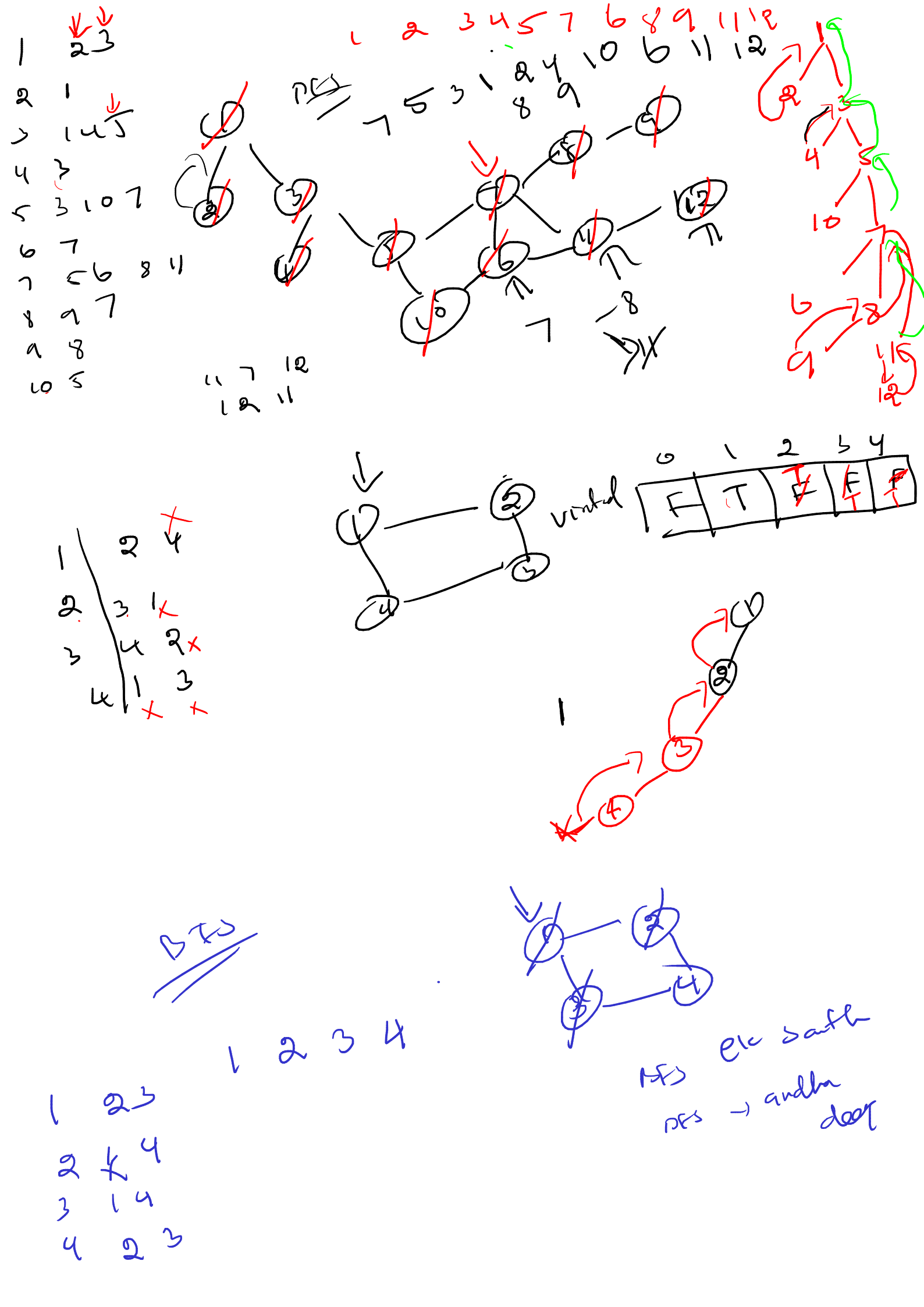


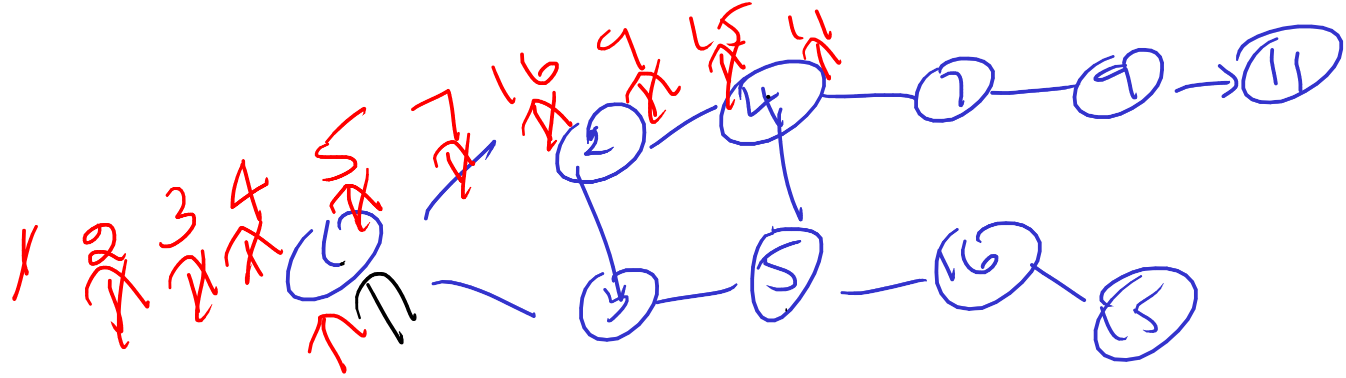


c1

c2

DFS(1) → 1 2 3 5 4 6


visited
0 1 2 3 4
F T F F F
DFS
BFS ek sath
DFS → andha
deep



1 2 3 4 5 7 16 9 15 11

queue

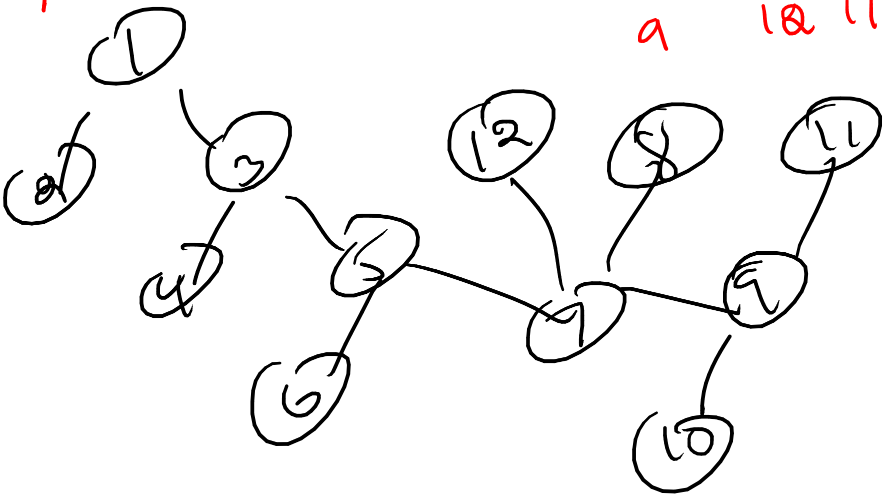
que

Ans

1 2 3 4 5 6 7 8

9 12 11 10

BFS



Selvam k

1) cycle graph

→ galaxy

march → 2020

WP

1 selvaraka

2) number islands

3) traffic origins

4) bipartite graph //

1) s → end { }

2) rotten edge

3) cycle det
   - undirected → DFS / BFS
   - directed → DFS

4) Bipartite

st	end
2	100

{2, 5, 100} ①

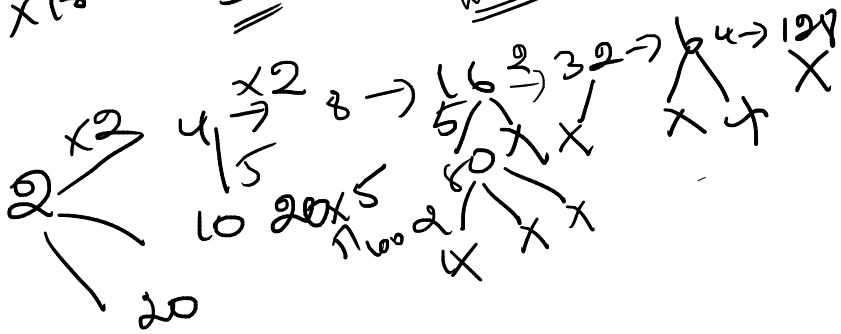
2 × 5 × 5 × 2 = 4 — minimal

2 + 5 × 10 = 3 — minimal

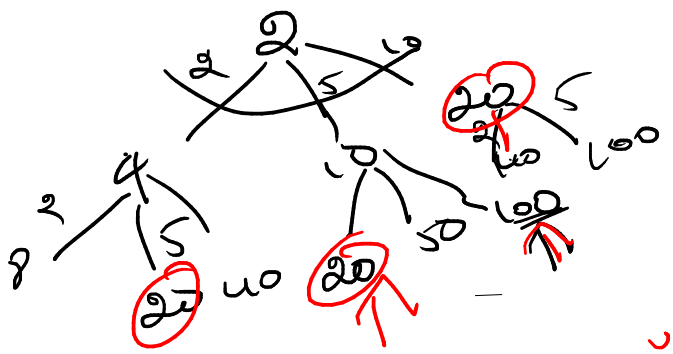
DFS

= 100 — 100 X

DFS



BFS



BFS → deep X

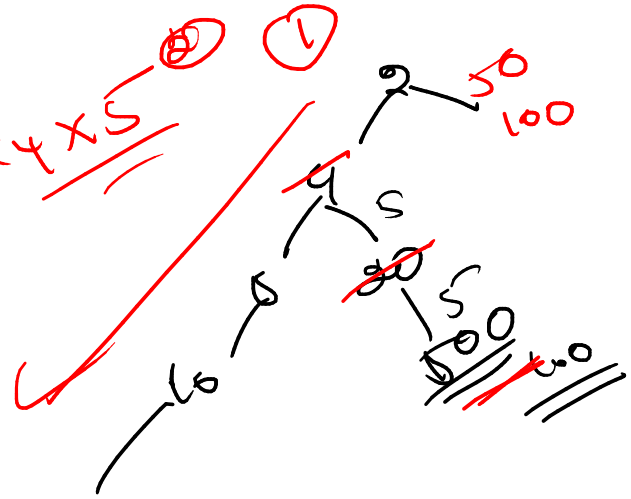
DFS → deep → rev

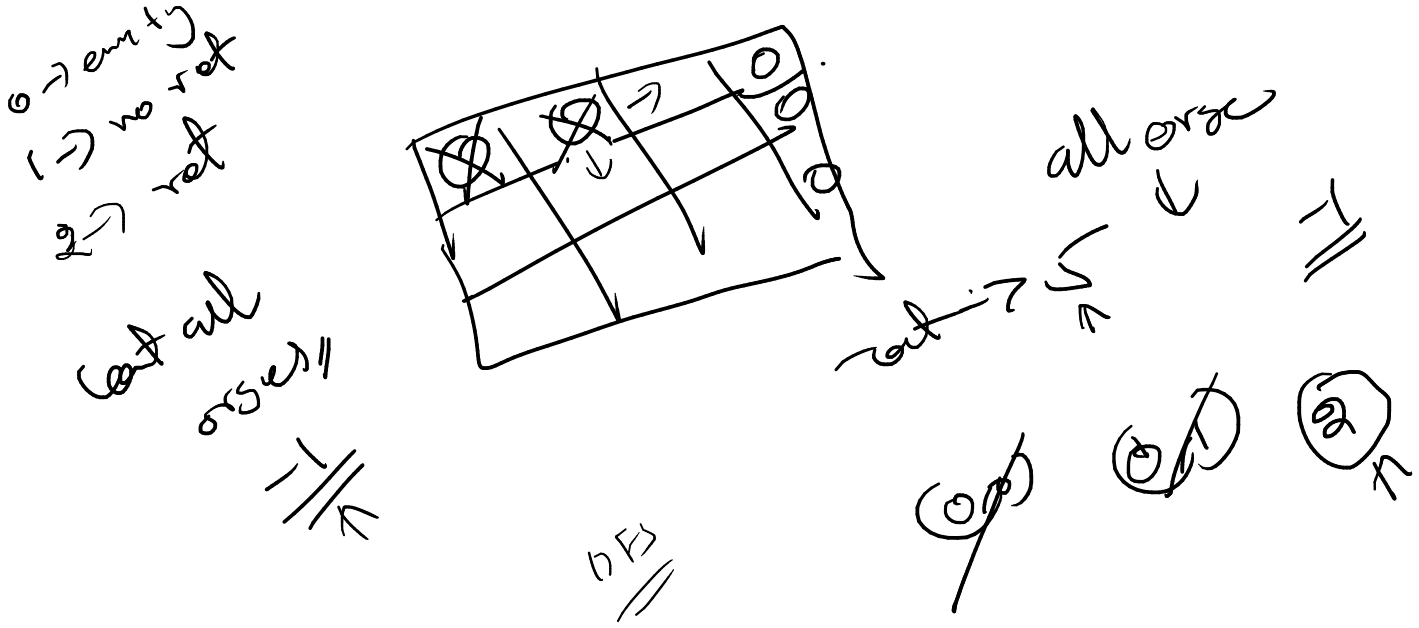
BFS → level

visited 100

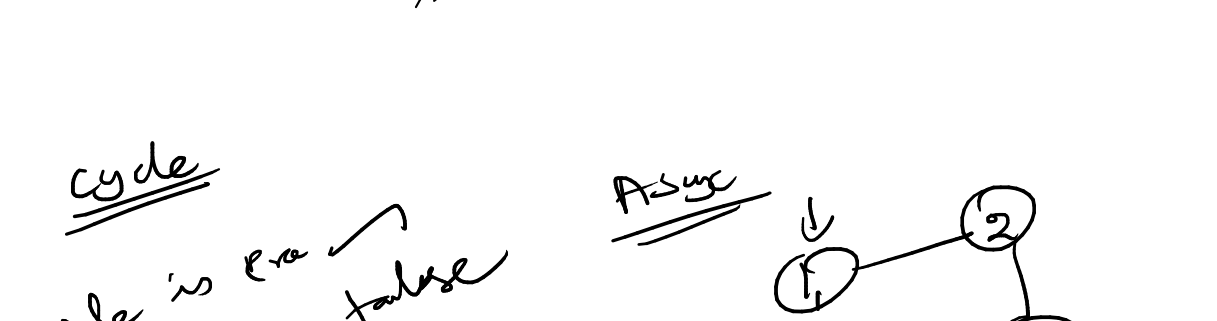
DFS

2 × 4 × 5





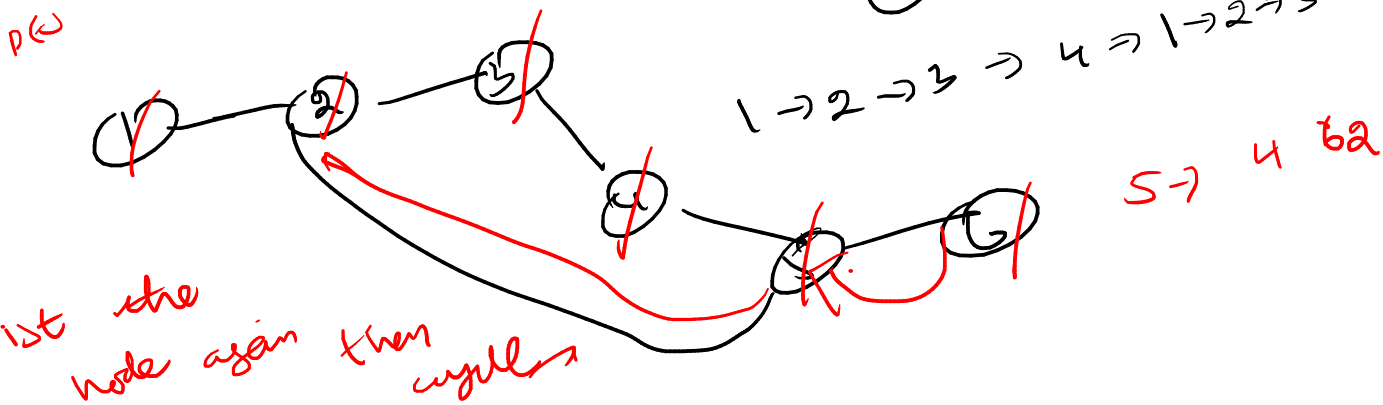
DFS



cycle

cycle is present → false

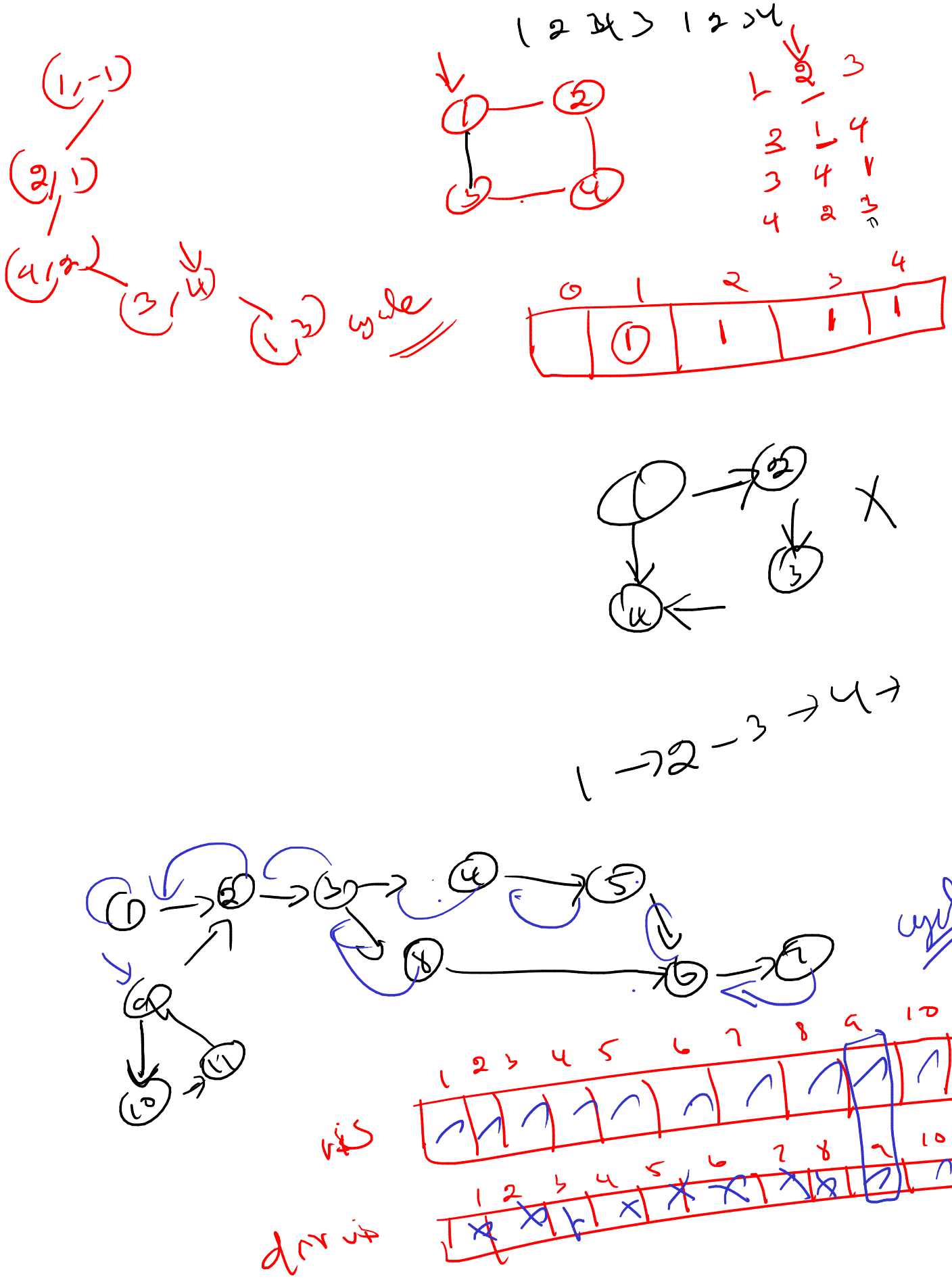
Asyc



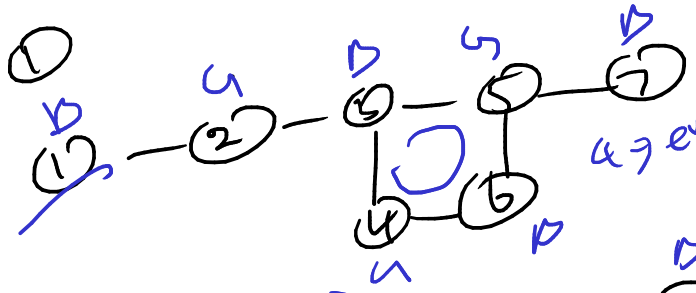
1 → 2 → 3 → 4 → 1 → 2 → 5

5 → 4 ka

revisit the node again then cycle


1 2 4 > 1 2 > 4
1 2 3
3 1 4
3 4 1
4 2 3
(1,-1)
(2,1)
(4,2)
(3,4)
(1,3) cycle
0 1 2 3 4
1 1 1 1
X
1 → 2 → 3 → 4 →
cycle
vis
dfs vis

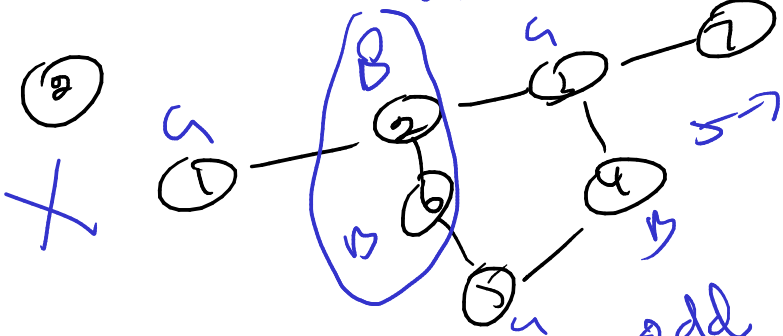
1) Bipartite
2) Topo sort
3) shortest path



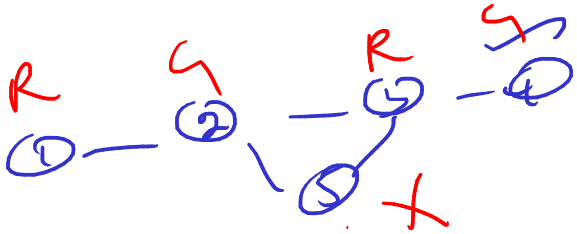
Bi cut

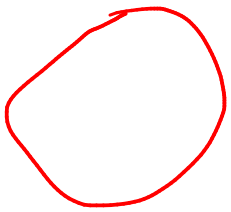
color whole graph with only 2 colors ↑

no two adjacent node should have same color

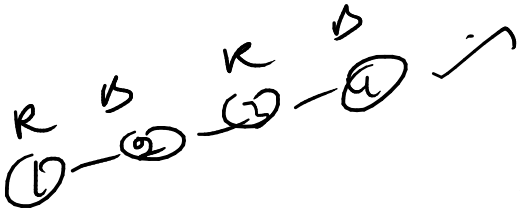


odd cycle X



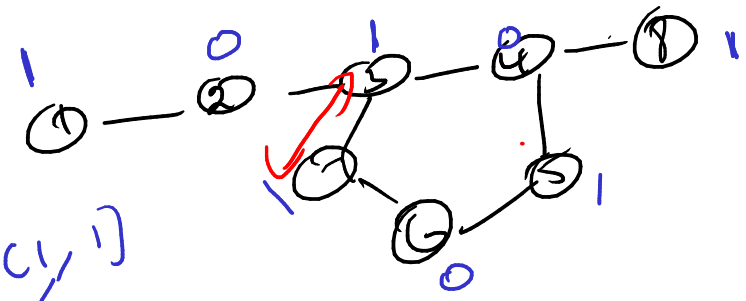


cycle length ⇒ ∴ odd ↷

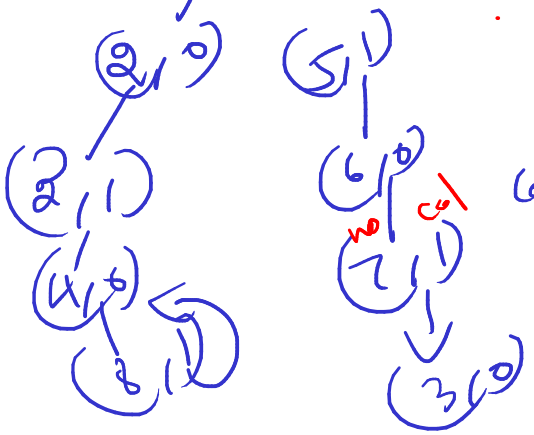


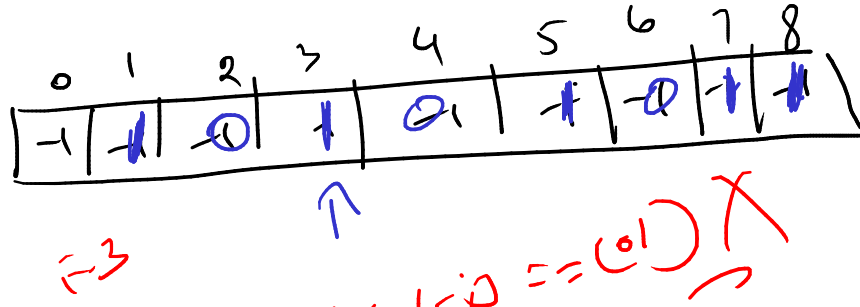
(cycle ∅ ∀ (ancy) % 2 = 1) X

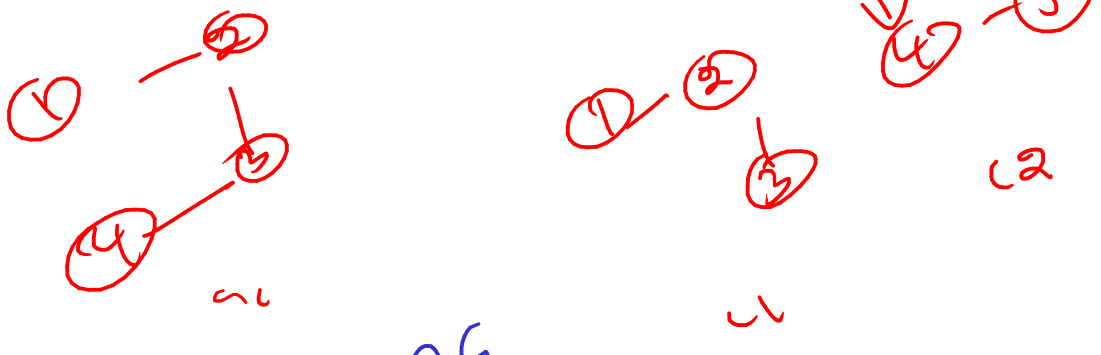
↗ else ↘



0 → R
1 → B
-1 → Not vl

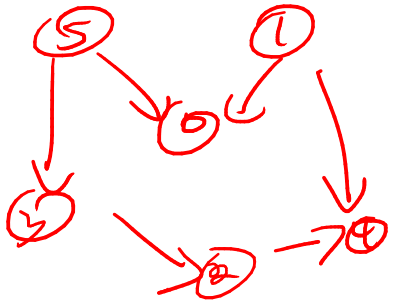






Topological sort DAG

Linear ordering of vertices



order 1 5 0 3 2 4

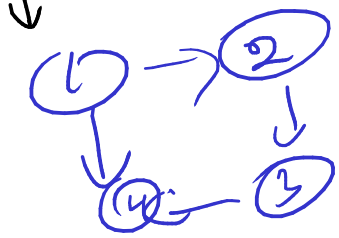
5 1 0 3 2 4

u → v → u should come before v

1) No cycle
2) Directed

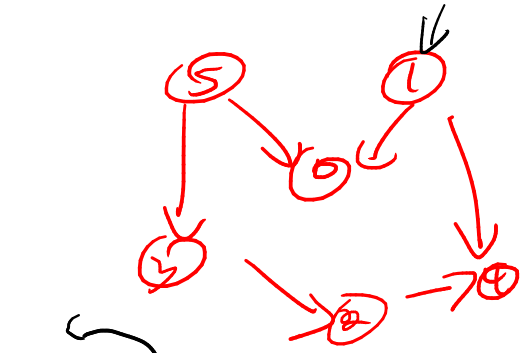
1 2 3 | 2

DAG

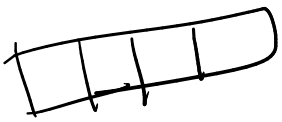


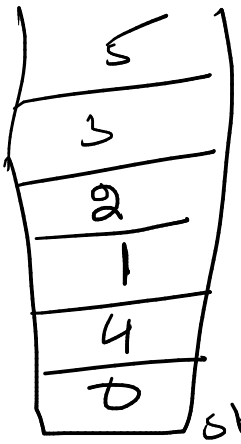
1 2 3 4

1 4 2 3



1) DFS
2) BFS → Kahn's Algo

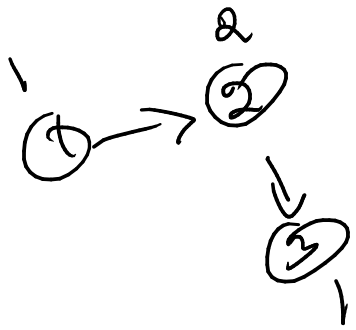




5 3 2 1 4 0

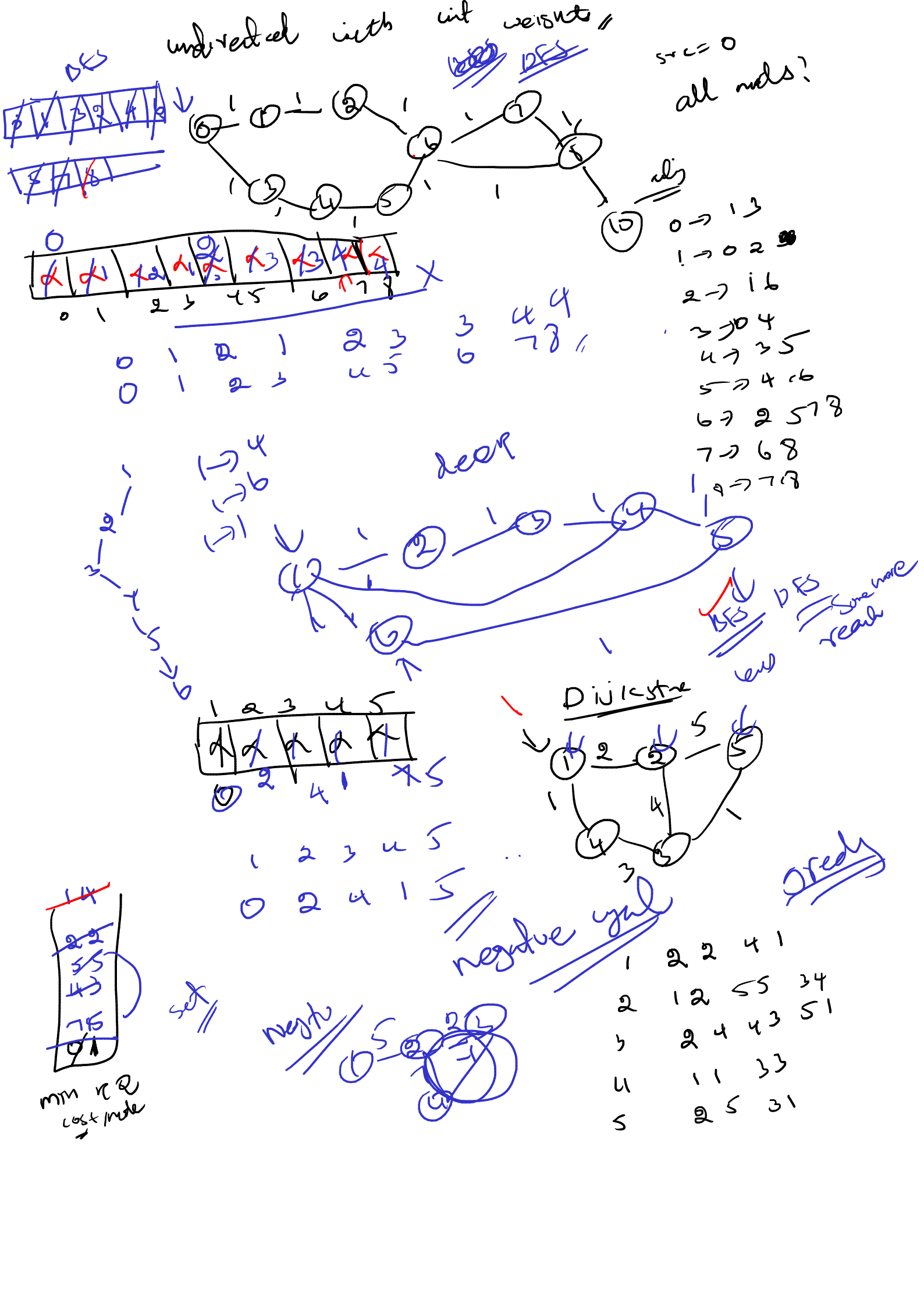
indeg 0 1 1

outdeg 1 1 0



BFS

1) Topo
2) Shortest
   ↳ undirected ①
   src → all nodes?
   BFS / ⓐ
   ↳ Dijkstra


undirected with int weights
BFS
DFS
src = 0
all nods?
0 → 1 3
1 → 0 2
2 → 1 6
3 → 0 4
4 → 3 5
5 → 4 6
6 → 2 5 7 8
7 → 6 8
8 → 7
deep
BFS DFS Same more reach
Dijkstra
negative cycle
greedy
set
next
min PQ
cost node