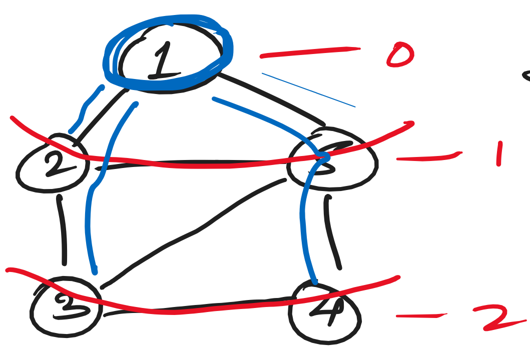
Graph Traversals : BFS & DFS

* Cycle Detection

* Breadth First Search
Level Order Traversal } * Multiple Answers



Starting node → (1) →

1, 2, 5, 3, 4

1, 5, 2, 4, 3

* Who are your neighbours?

Output : → 1, 2, 5, 3, 4

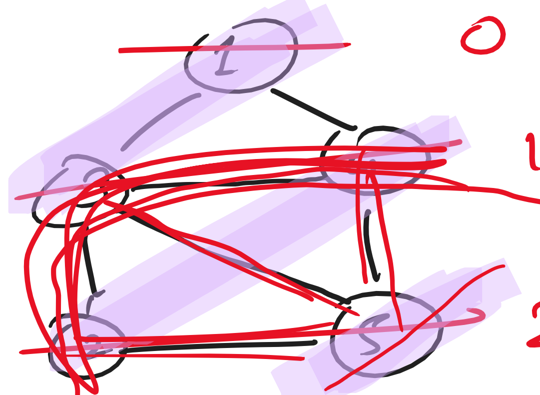
Adj List :

1 → 2, 5
2 → 1, 3, 5
3 → 2, 4, 5
4 → 3, 5
5 → 1, 2, 3, (4)

Visited

1 → ~~F~~ 1
2 → ~~F~~ 1
3 → ~~F~~ 1
4 → ~~F~~ 1
5 → ~~F~~ 1

queue



BFS :

Starting node → 1
Starting node → 5

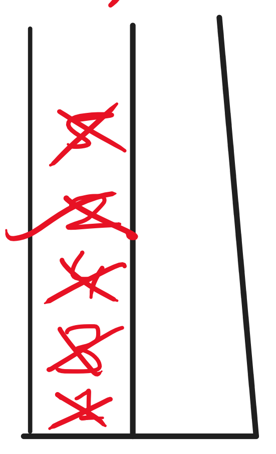
2 (1) → 1 3 4 2 5
(5) → 5 2 4 3 1 / 5 4 2 1 3

Adjacency list :

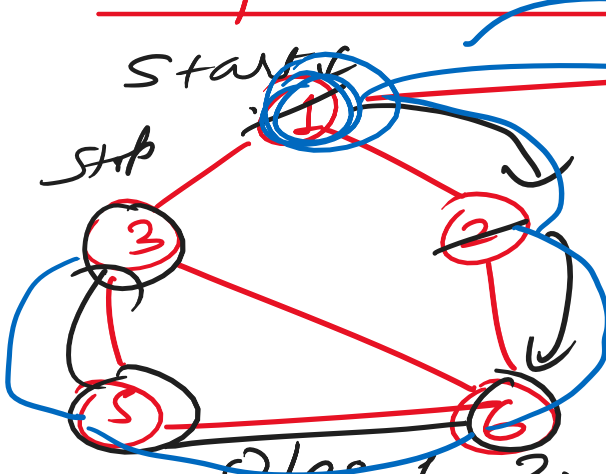
node — it it it
1 → 3, 4
2 → 3, 5
3 → ~~1~~, 2, 4, 5
4 → 1, 3, 5
5 → 2, 3, 4

Visited

1 → ~~F~~ T | F
2 → ~~F~~ T | F
3 → ~~F~~ T | F
4 → ~~F~~ T | F
5 → ~~F~~ T | F



Depth First Search Technique :



o/p : 1, 2, 6, 5, 3

* The o/p depends on where you traverse from the starting node.

dfs(1) → dfs(2), ~~dfs(3)~~, dfs(4)
dfs(2) → dfs(6) → dfs(5) → dfs(3)
dfs(4) → dfs(7)

Adjacency List : →

1 → (2), (3), (4)
2 → 1, (6)
3 → 1, 5, 6
4 → 1, 7
5 → 3, 6
6 → ~~2~~, 3, 5, 4
7 →

Visited 1, 3, 5, 6, 2, 4, 7

~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1
1	2	3	4	5	6	7

Advanced Graph :

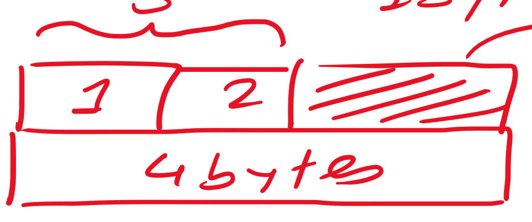
Cycle Detection →

TUF
Kunal
Love Babbar
Telusko
Code With Mosh

Greedy Algorithms : → TJIT

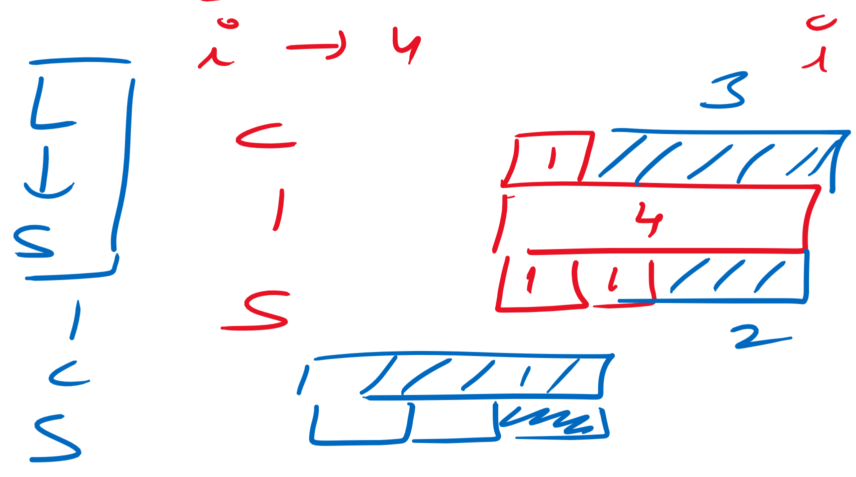
* Padding & Greedy Alignment

c → 1
s → 2
i → 4



4 + 4 = 8 bytes

4 + 4 + 4 = 12 bytes



* Minimum number of Coins :

arr = {1, 2, 5, 10, 20, 50, 100, 500, 2000}

Value = 31

20 + 10 + 1 = 3

31 − 20 = 11 — (1)
11 − 10 = 1 — (2)
1 − 1 = 0 — (3)