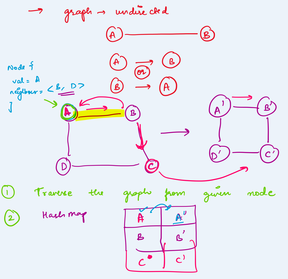
## Q1 Clone graph

→ graph → undirected




① Traverse the graph from given node

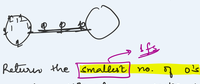
② Hash map

| | |
|---|---|
| A | A' |
| B | B' |
| C | C' |

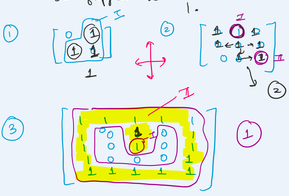
## Q2 Shortest Bridge

2D binary array
( 0 and 1 )

2 islands



Return the smallest no. of 0's that must be flipped to make both islands as 1.

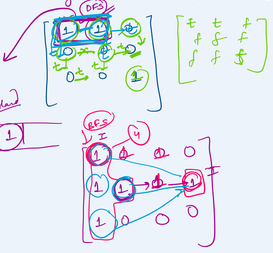



① We can use dfs/bfs to identify and find the islands.

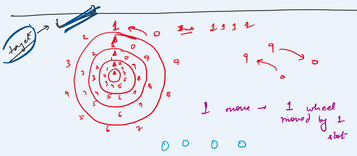
② start bfs from any 1 of the island and keep traversing until you find another 1's of the other island.

* Level at which we find other island, is going to the shortest path which can connect 2 islands.

* Also, min flips required to create a bridge between the two islands.




---




① deadends →
[0201, 0101, 0102, 1212, 2002]

target → 0202

I II III IV
0000
↓
1000
↓
1100
↓
1200
↓
1201
↓
1202
↓
0202

0000
↓
0001
↓
0002
↓
0002 — Invalid
↓
0202
↓

---

find minimum no. of turns required to open the lock.

→ shortest path problem

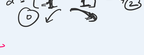
→ 10,000 nodes [0000 → 9999]

0000 0001 0002 0213 0100

→ edge b/w nodes if the digits differ by 1.

→ both nodes are not deadends.

BFS



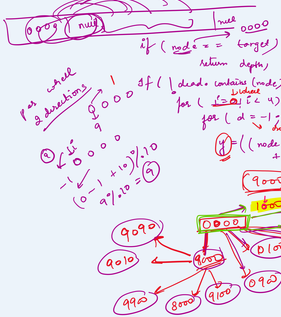

① Create a set and add all deadends to it

② Create a queue and add root of the graph to the queue ('0000')

③ Create another hashset to keep track of visited nodes.

④ while !!queue.isEmpty{

if (node = null)
depth ++;

if ( node = = target)
return depth;

for ( int i = 0; i < 4; i++)
for ( d = -1; d <= 1; d++ = 2)
if ( (node.charAt(i) - '0') + d + 10) % 10 )