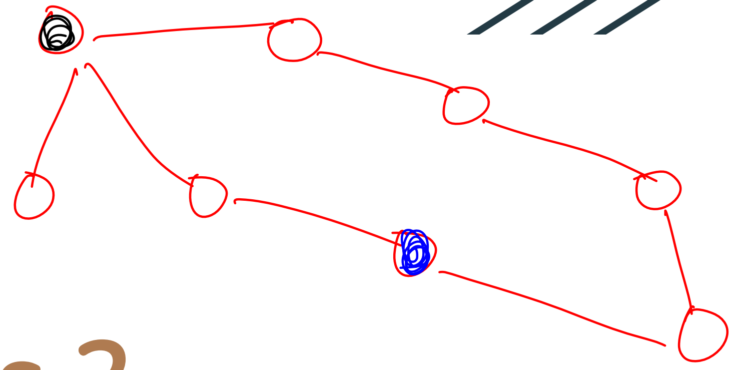


Dfs

Graphs 2

Traversals + Bipartite Graphs

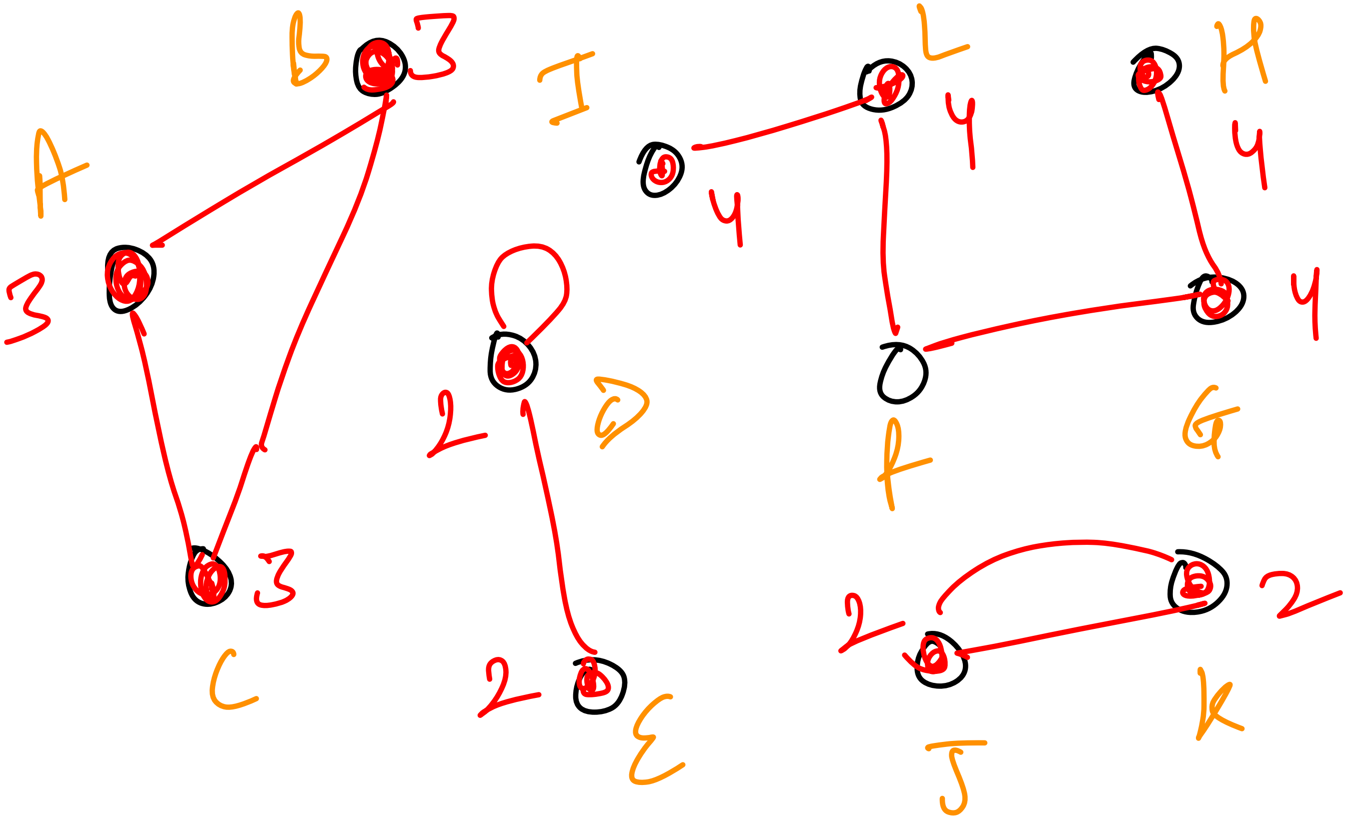
Bfs — single source shortest path algorithm for an unweighted graph

-Priyansh Agarwal

Problems on Traversals

- ✓ Find out the size of each connected component in the graph
- ✓ Find out the maximum area of an island on a binary matrix
- ✓ Minimum time to rot all oranges problem
- ✓ ~~Distance of nearest cell having~~ 1 [Multi source BFS]

find out the nearest hospital in a city to every house



$1 \leq n \leq 10^5$

$1 \leq m \leq 10^6$

$[v, e]$

$[n, m]$

for every node find out the size of the connected component of which that node is a part

```
void  dfs (curr, edges, visited, vector<int> comp){
    comp.push_back(curr);
    visited[curr] = true;
  for( neighbour : edges[curr]){
      if(!visited(neighbour]){
          dfs(neighbour, edges, visited, comp);
      }
}
vector<bool> vis(n,false);
for (int i=0; i<n; i++){
    if(!visited[i]){
            vector <int> component;
            dfs(i, edges, visited, component);
            for ( node: component)
                    ans(node] = component.size()
        }
}
```

$$\underline{O(n)} + O\left(\underline{\sum_{i=0}^{n} \underline{N(i)}}\right)$$

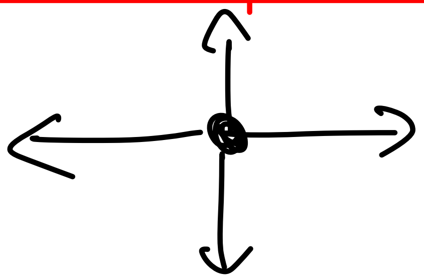
$$\underline{O(n+m)}$$

② find out the maximum area of 1s in a binary matrix of n x m

0	1	1	0	0
1	1	0	1	1
0	1	1	1	1
0	1	0	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0

every cell is 1x1

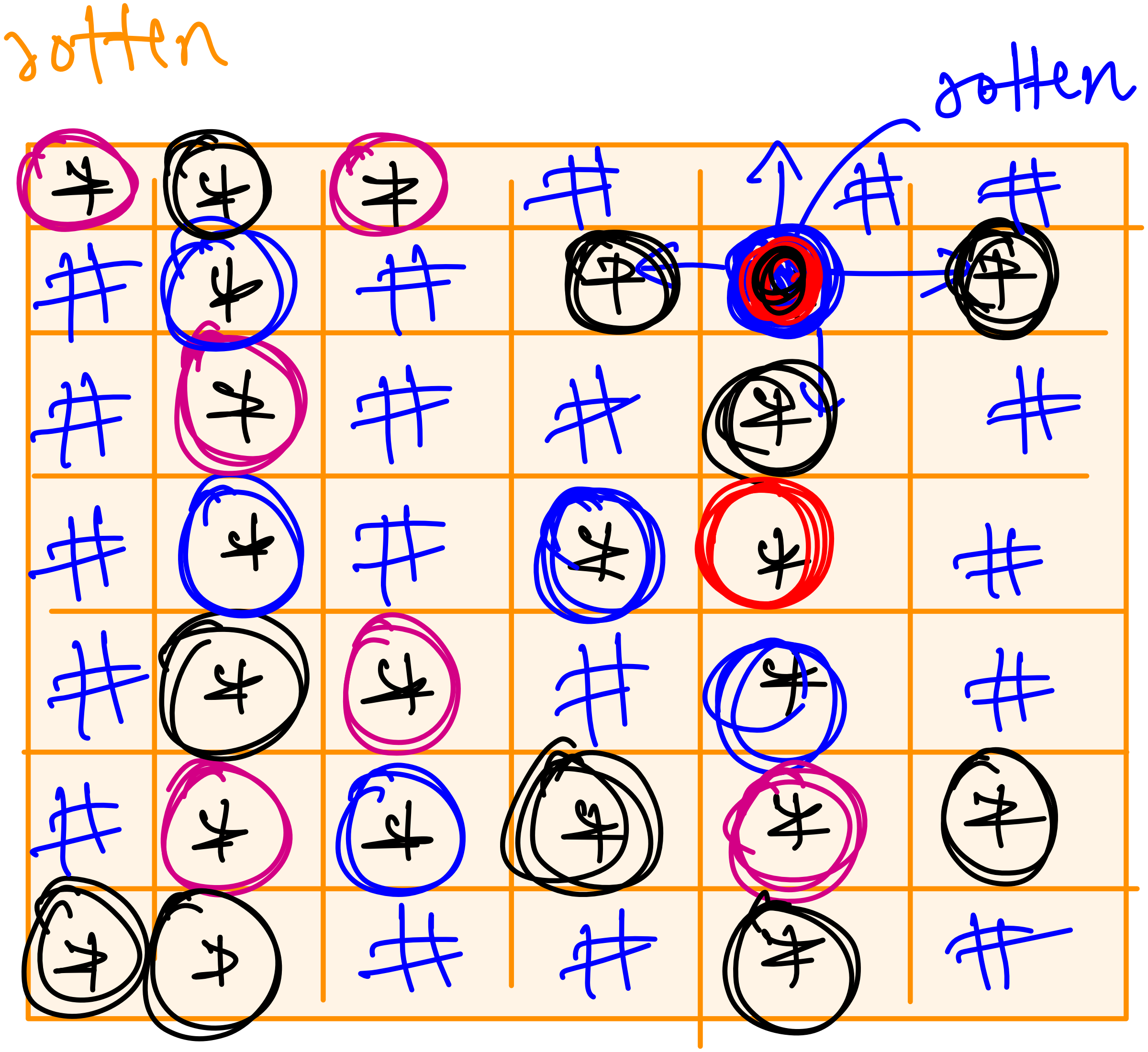
$1 \le n, m \le 10^3$



③ minimum time to rot all oranges in a matrix when one of the oranges is rotten

n, m

$1 \leq n, m \leq 10^3$



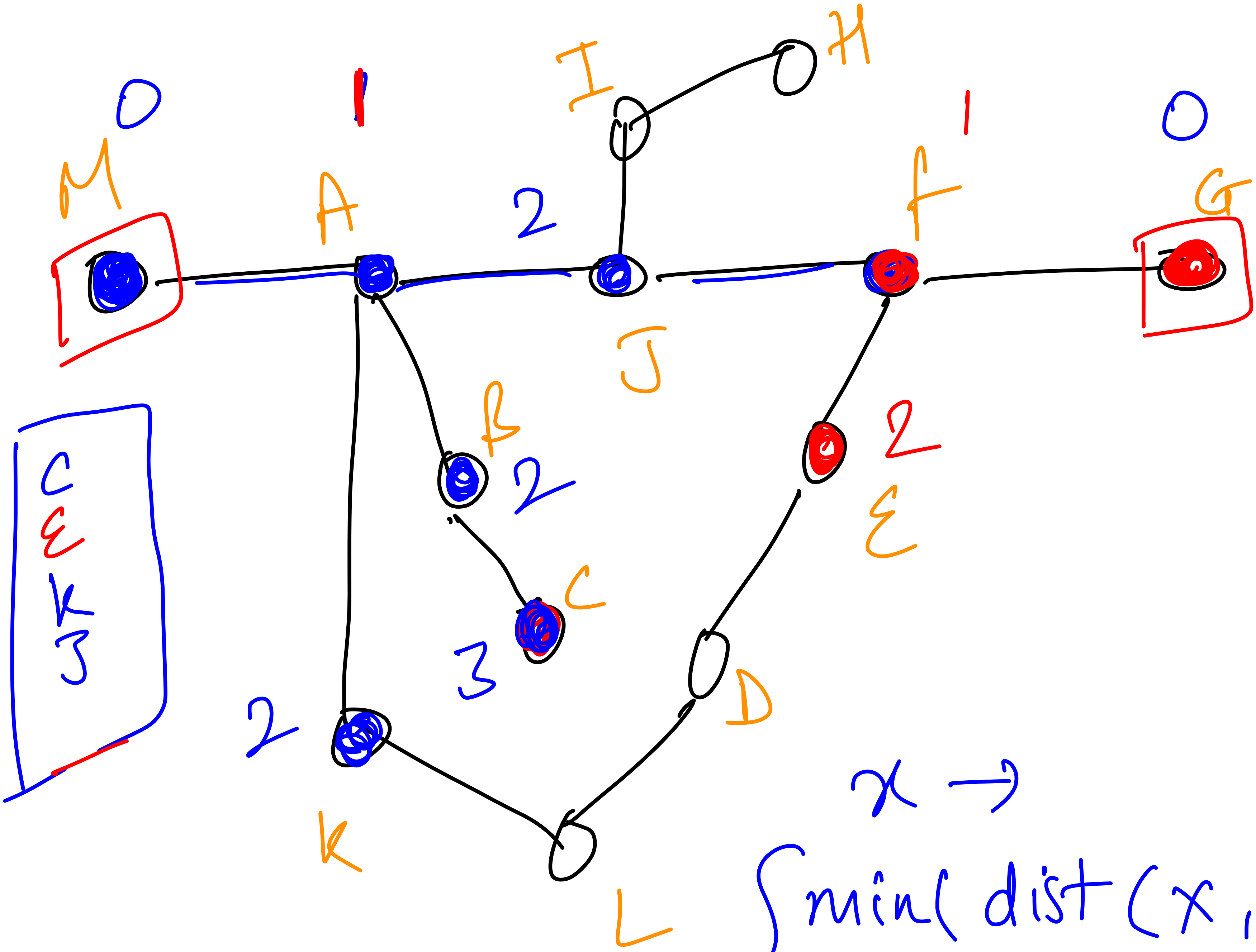
*	*	*	#	#	#
#	*	#	*	*	*
#	*	#	#	*	#
#	*	#	*	*	#
#	*	*	#	*	#
#	*	*	*	*	*
*	*	#	#	*	#

entire set of oranges in 1 component

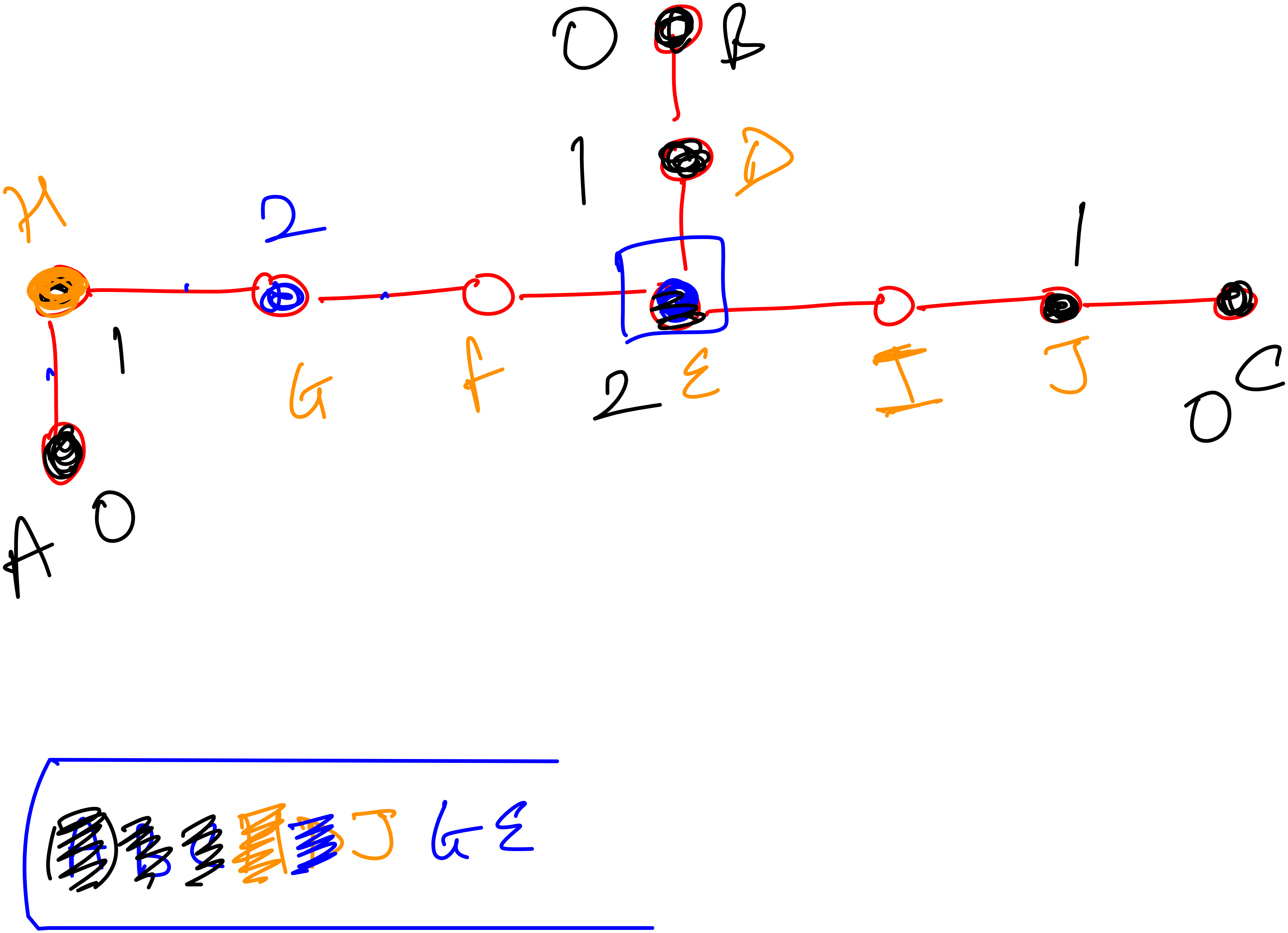
(4) You are given a city in the form of a matrix where every cell is either a house, hospital or empty patch. for each house find out the distance of that house to the nearest hospital

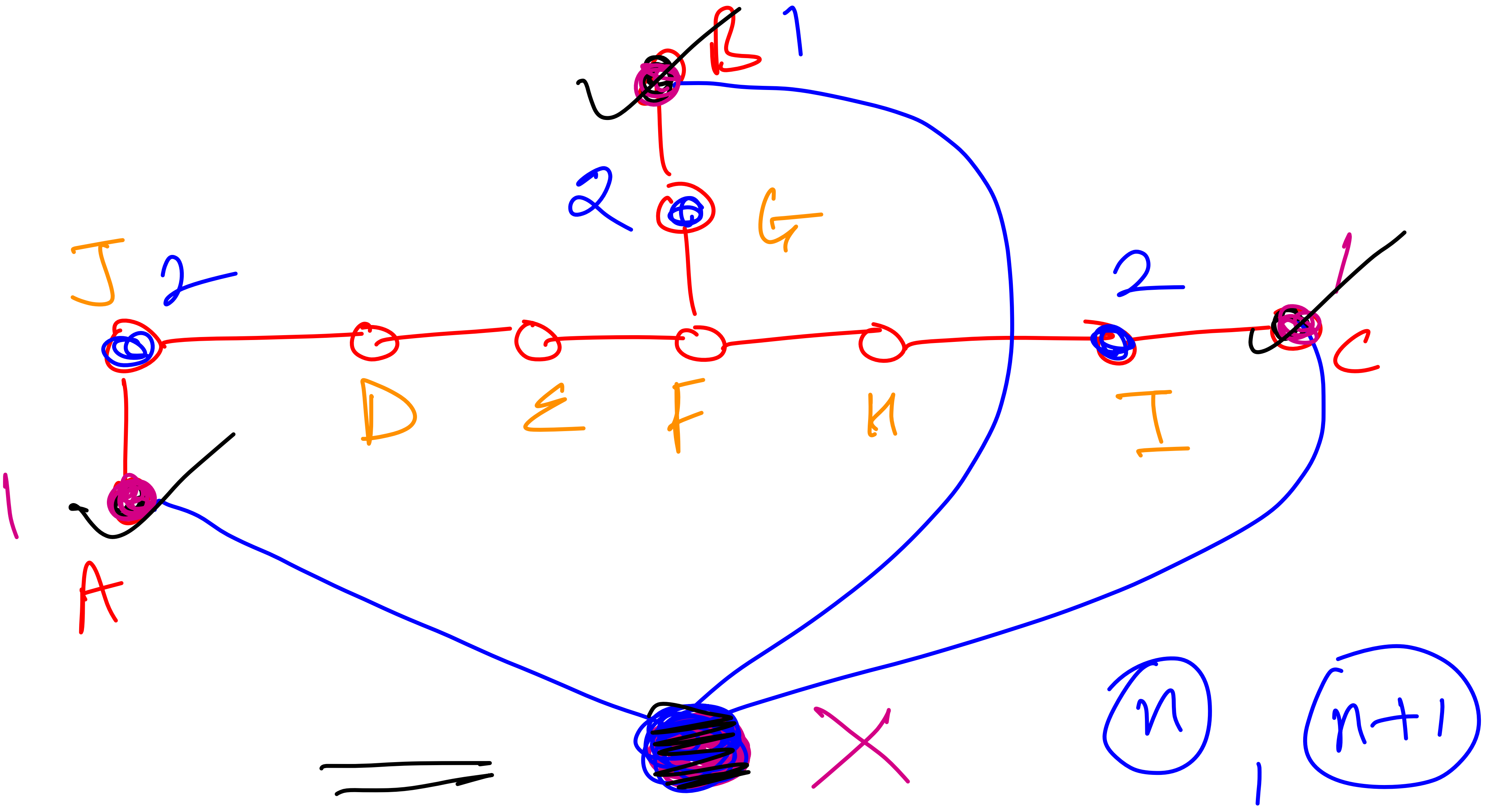
$1 \le n, m \le 10^3$

🟠	—	—	—	—	—	—	🟠
—	—	🟠	—	🟠	🔴	—	—
—	—	🔴	—	—	—	🟠	—
—	🟠	—	—	🟠	—	—	—
🔴	—	—	🔴	—	—	🔴	🔴
—	🔴	🟠	—	—	—	🟠	—
—	—	—	—	—	🟠	—	—



$$x \to \left\{ \min(\mathrm{dist}(x,G), \mathrm{dist}(x,M)) \right.$$

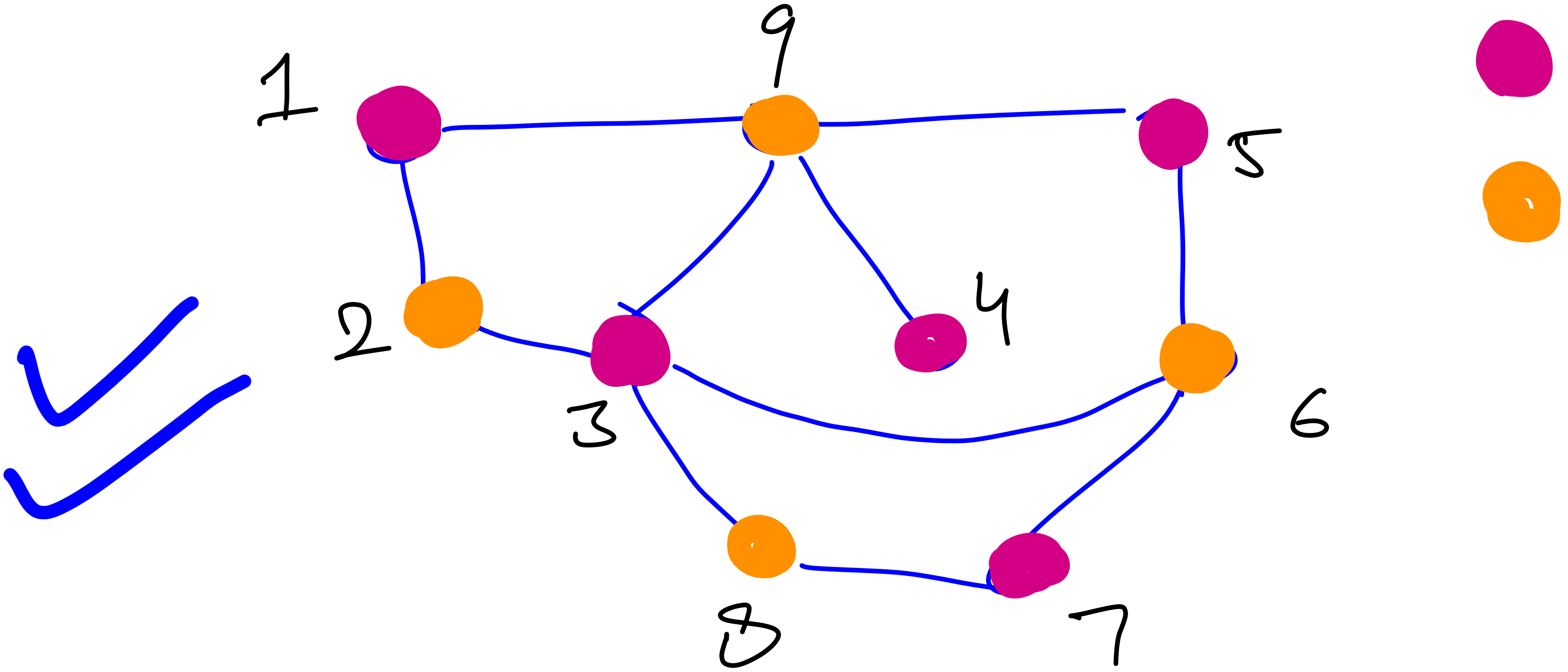
0
B
1
D
H
2
1
1
G
F
2
E
I
J
0
C
A
0
J
G
E

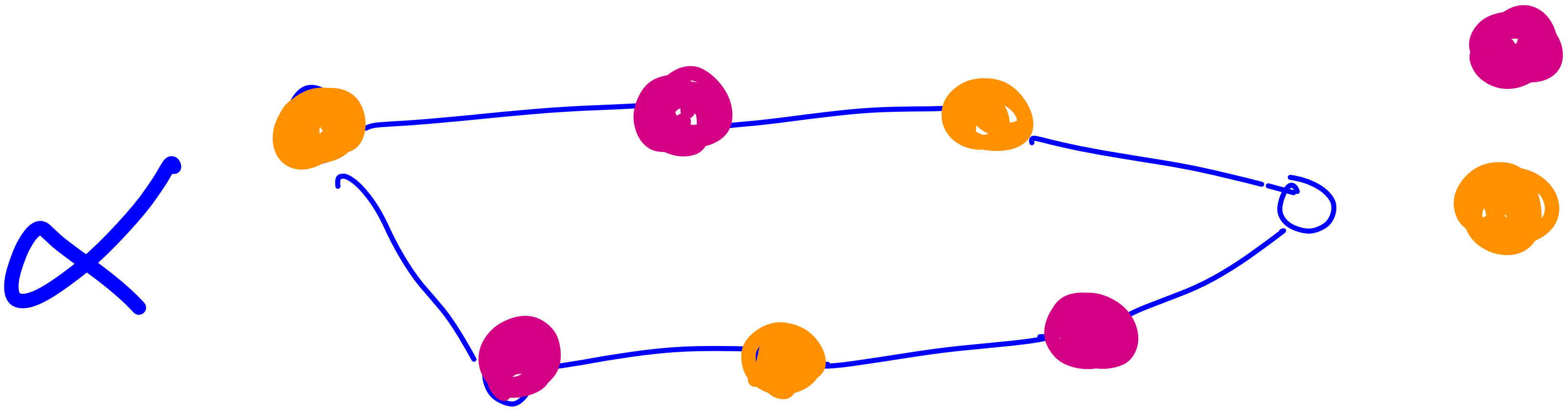

B
1
2
G
J
2
D
E
F
H
2
I
C
1
1
A
=
X
n
,
n+1

Bi-partite Graphs

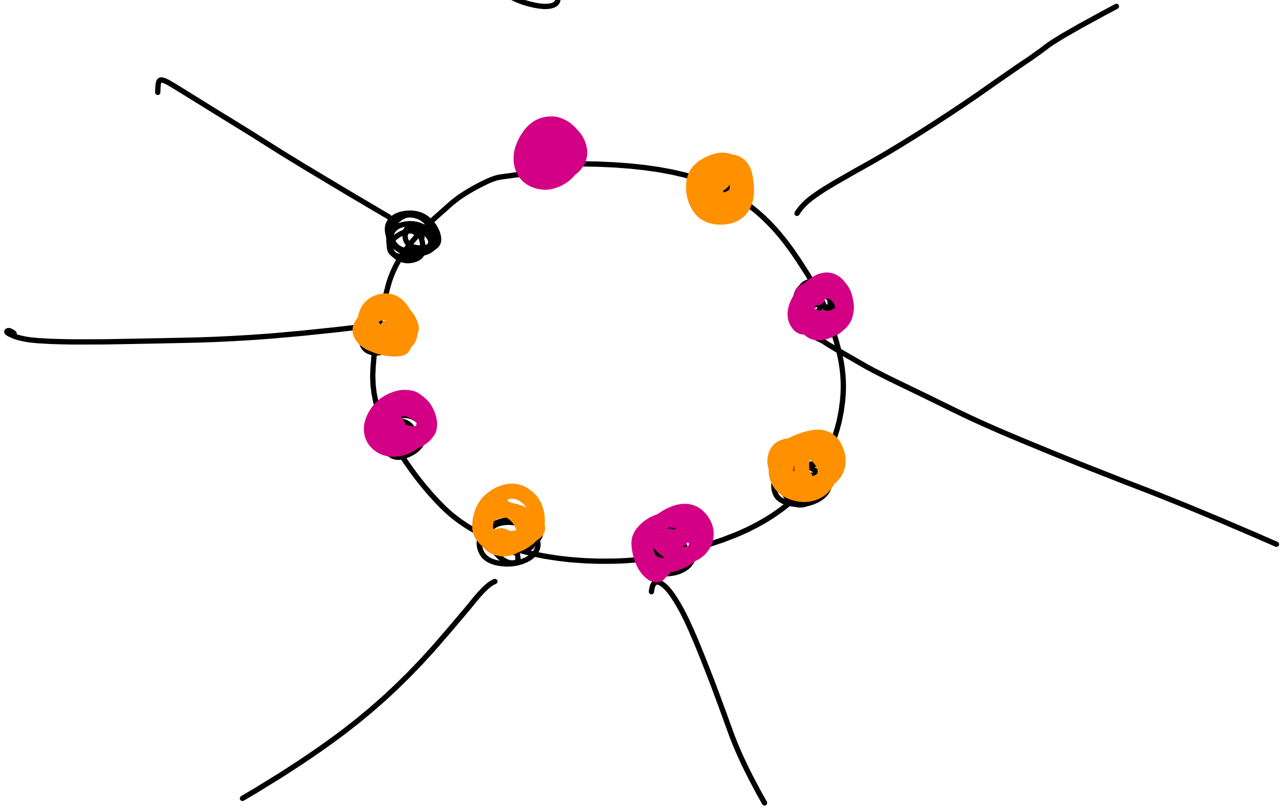
- ✓ Definition
- Algorithm
- ✓ Odd Length Cycle
- Tree Property
- Problem: Link
- ICPC World Finals problem Link

A graph for which you can color all the nodes with one of the two colors such that no two neighbours have the same color

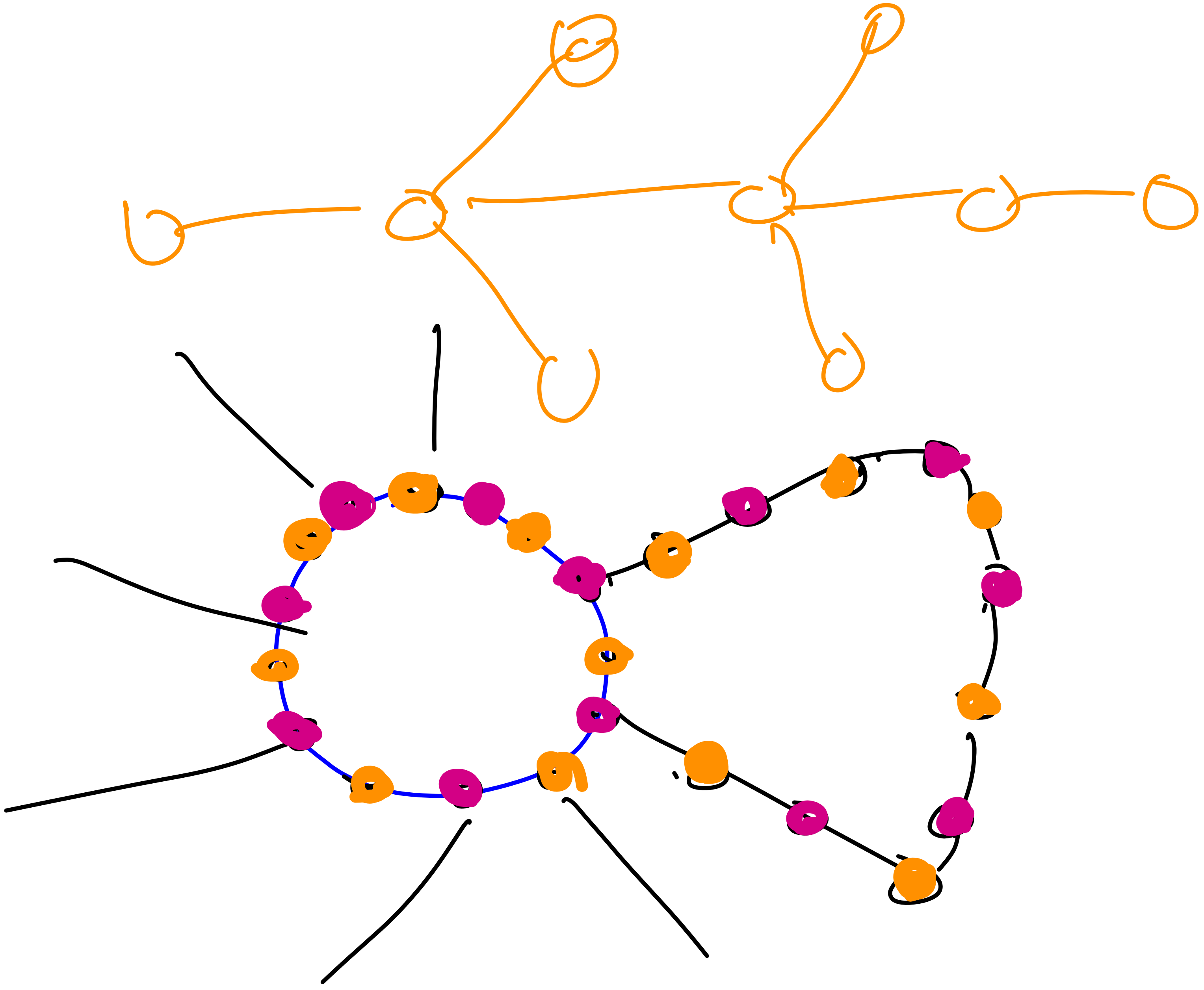

✓
1
9
5
2
4
3
6
8
7


✗

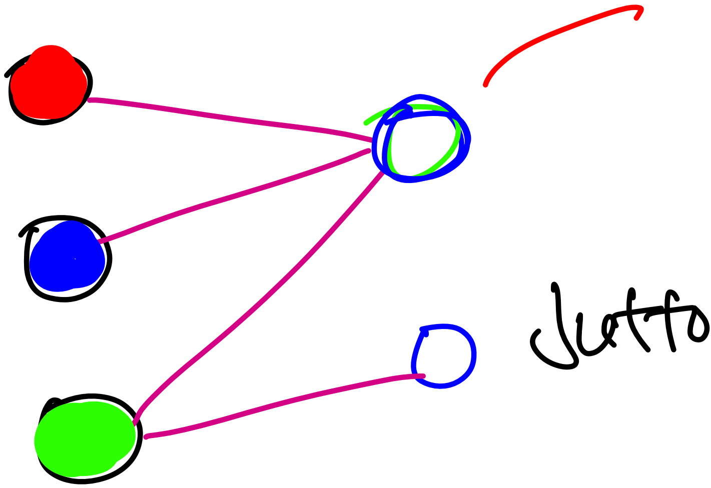
A graph is bipartite if and only if it doesn't contain an odd length cycle.



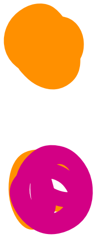
if there are no cycles then it is # obvious

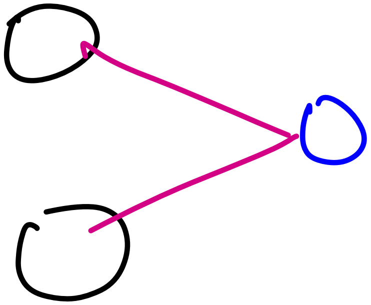


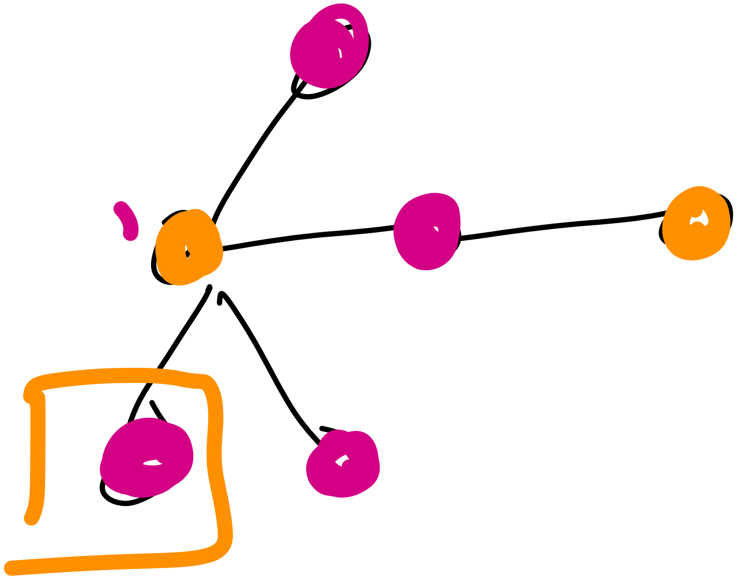
Bulbs

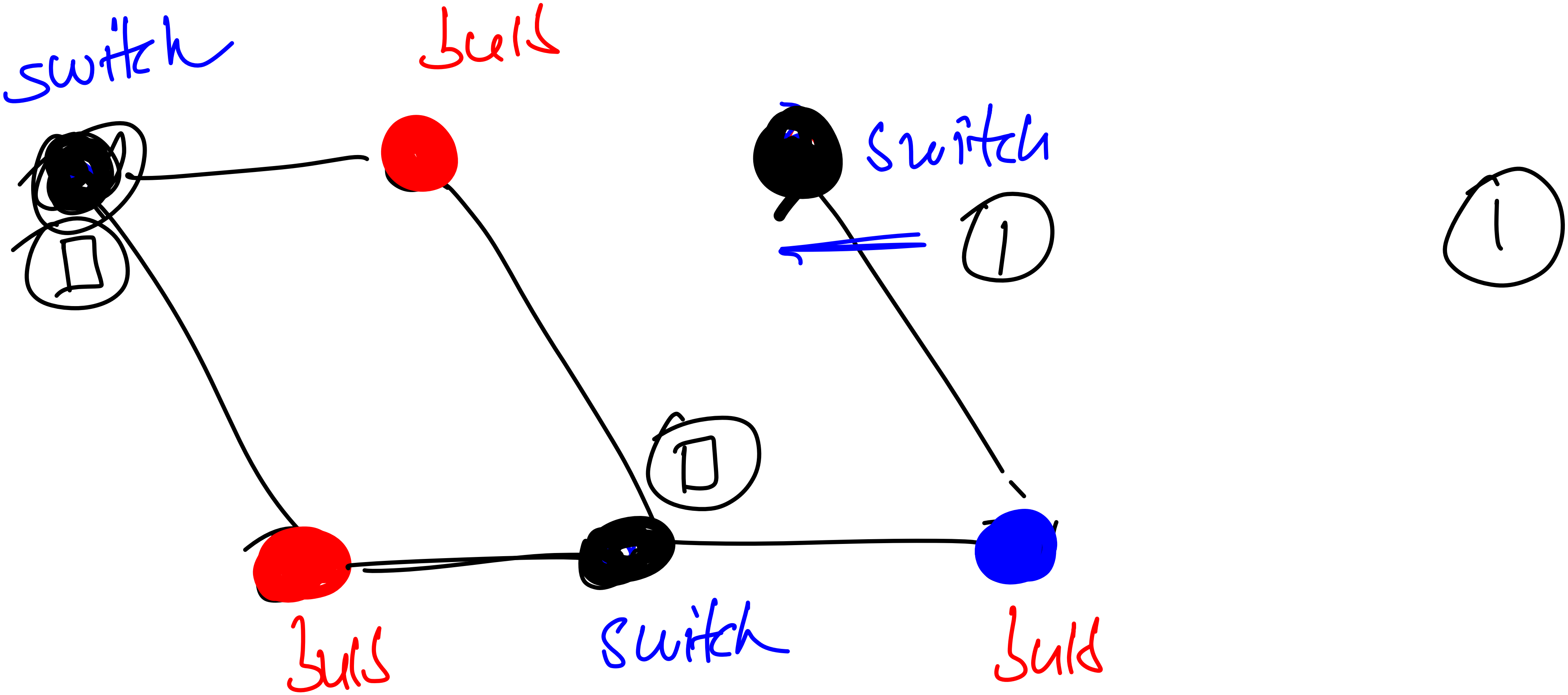


buttons | switches







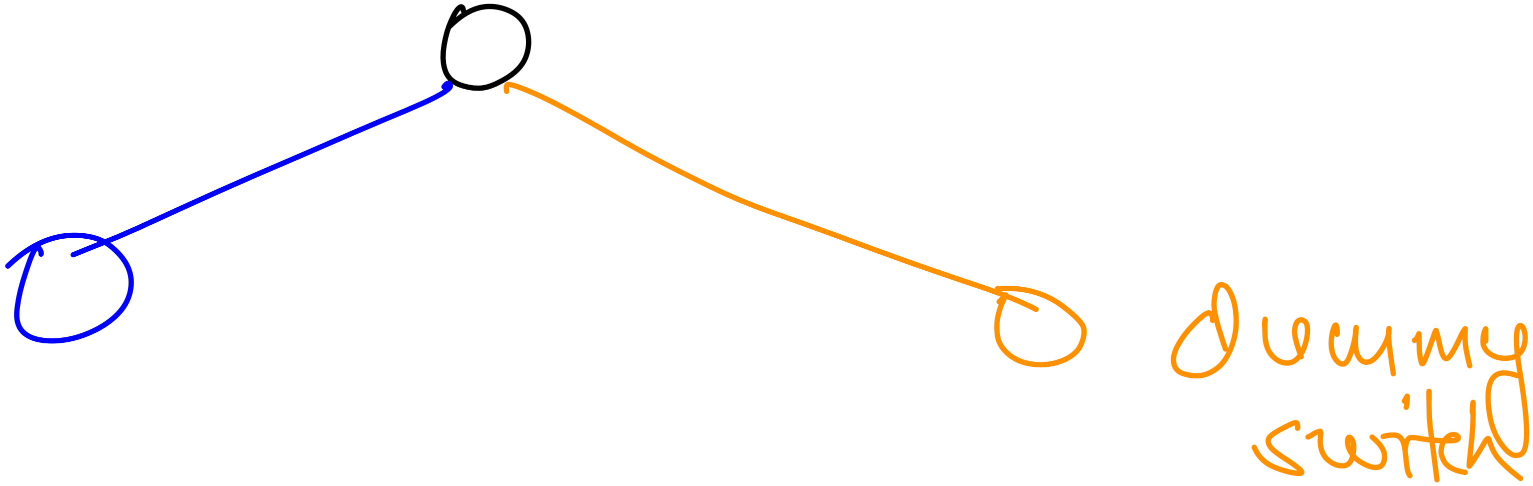

switch
bulb
0
switch
1
1
0
bulb
switch
bulb

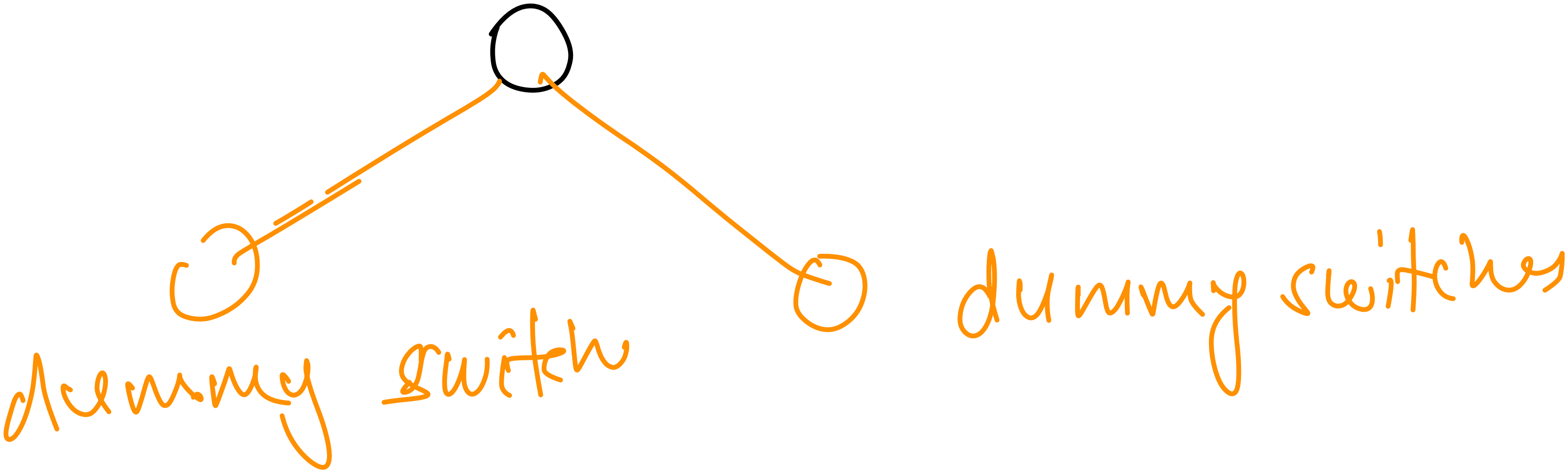
multiple components

→ for each try out 3 possibilities

→ some bulbs being connected to just 1 switch

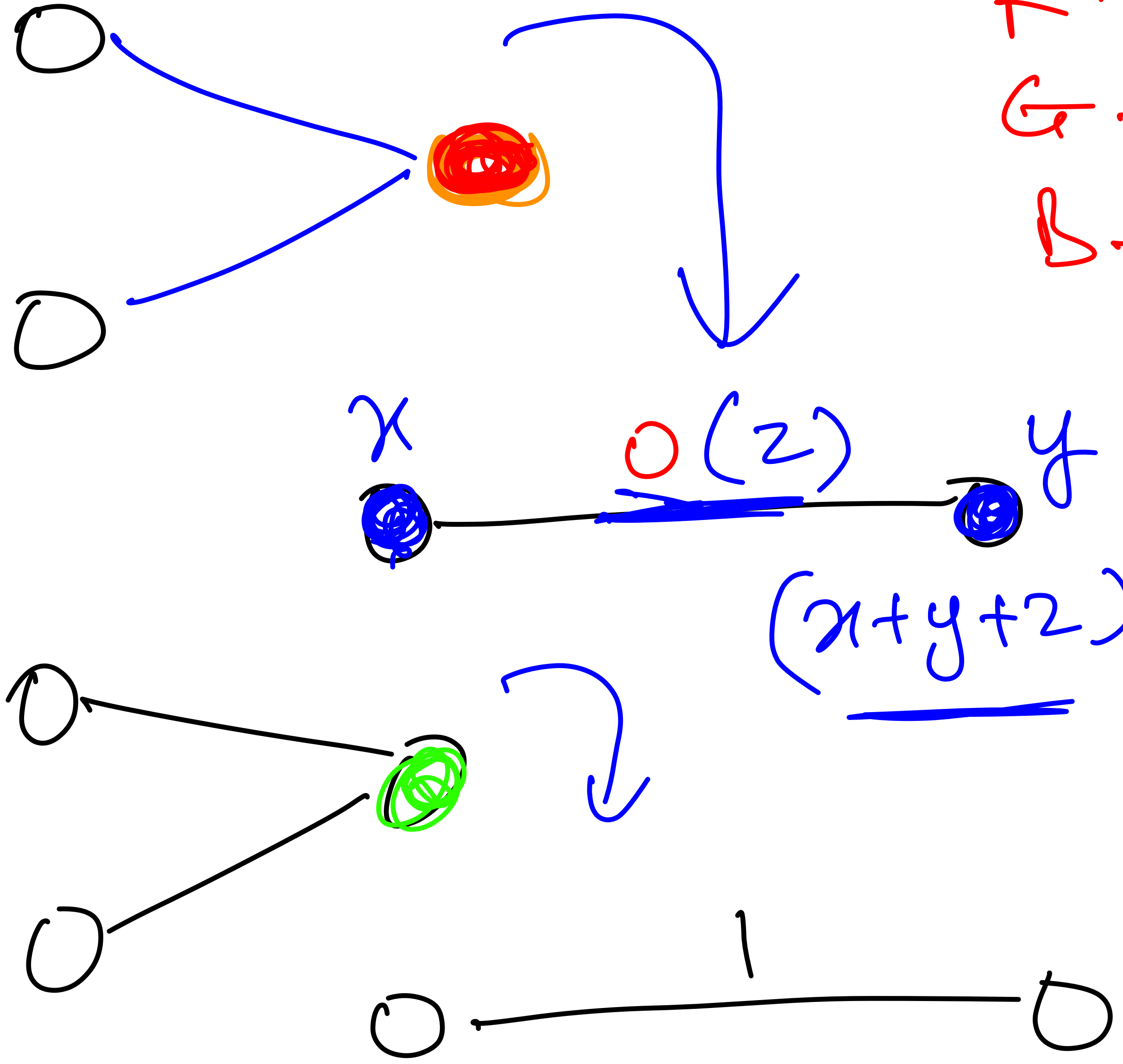
→ some bulbs not being connected


dummy switch


dummy switch
dummy switches

```
vector <bool> vis(m, false)      X
for ( int i=0 ; i< m ; i++)
        if( !vis[i]) {
                ↳ 0 ———— ar
                ↳ 1 ————
                ↳ 2 ————
                X += min( ———— )
        }
}
```

R → 0
G → 1
B → 2



$$\underline{(x+y+2)\%3=0}$$