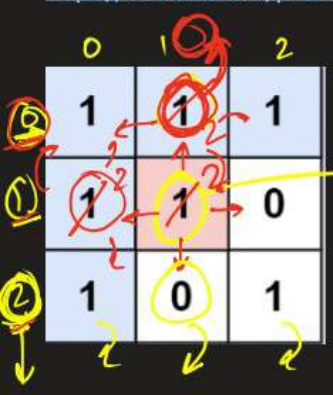
# Trees & Graphs Lecture 4

Tuesday, 20 August 2024 6:06 AM

https://leetcode.com/problems/flood-fill/




init = 1

pt[0] > 0

sr, sc = (1, 1)

color = 2

n = 3

q = appendleft → pop

(0,2) (0,0) (1,0)

pt = (1,1) pt = (0,1) (1,0)

pt[0] > 0 and ans[pt[0]-1][pt[1]] == init

pt[0] < n-1

```python
from collections import deque

class Solution:
    def floodFill(self, image: List[List[int]], sr: int, sc: int, color: int) -> List[List
[int]]:
        ans = image
        n = len(image)
        m = len(image[0])
        init = image[sr][sc]
        q = deque()
        if init != color:
            q.appendleft((sr, sc))
        while(q):
            pt = q.pop()
            ans[pt[0]][pt[1]] = color
            if pt[0]>0 and ans[pt[0]-1][pt[1]] == init:
                q.appendleft((pt[0]-1, pt[1]))
            if pt[0]<n-1 and ans[pt[0]+1][pt[1]] == init:
                q.appendleft((pt[0]+1, pt[1]))
            if pt[1]>0 and ans[pt[0]][pt[1]-1] == init:
                q.appendleft((pt[0], pt[1]-1))
            if pt[1]<m-1 and ans[pt[0]][pt[1]+1] == init:
                q.appendleft((pt[0], pt[1]+1))
        return ans
```

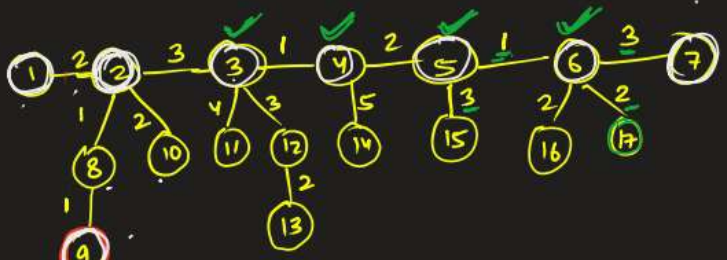
https://www.geeksforgeeks.org/problems/depth-first-traversal-for-a-graph/1

```cpp
// } Driver Code Ends
class Solution {
  public:
    void dfs(int p, vector<int> adj[], vector<int> &ans, vector<bool> &vis) {
        ans.push_back(p);
        vis[p] = true;
        for(int i: adj[p])
            if(!vis[i])
                dfs(i, adj, ans, vis);
    }

    // Function to return a list containing the DFS traversal of the graph.
    vector<int> dfsOfGraph(int V, vector<int> adj[]) {
        vector<int> ans;
        vector<bool> vis(V, false);
        dfs(0, adj, ans, vis);
        return ans;
    }
};
```

$T = O(V+E)$
$S = O(V)$

https://codeforces.com/contest/1004/problem/E




1-7 ✓
10-7 ✓
9-7 ✓
13-7 ✓

n = 17
e = 16
undirected

9 - 16 (11)
1 - 16 (11)

at most k shops
k = 4
r = 4

$f_1 = 8,\ f_2 = 2,\ f_3 = 3,\ f_4 = 11$

a = 5

objective :- minimize

$\max_{a} \min_{1 \le i \le r} d_{a,f_i}$

8, 2, 3, 11
a = 7 d = 7

a = 5 → 3
a = 7 → 7

a = 17 → 6
a = 1 → 2
a = 9 → 1

a = 8 (0)

Shops:- 3, 4, 5, 6 → [5]

ans = [5]

Diameter:- path on the tree with maximum distance

① Find the diameter of the tree.

Task
- From any node, find the farthest node (a)
- From a, find the farthest node (b)
- [a-b] is a diameter

```cpp
umap<int, int> a[100001];

void solve() {
    int n,k,u,v,d;
    cin >> n >> k;
    FOR(i, 0, n-1) {
        cin >> u >> v >> d;
        a[u][v] = d;
        a[v][u] = d;
    }
}
```

in.txt

```
17 5
1 2 2
2 3 3
2 8 1
2 10 2
8 9 1
3 11 4
3 12 3
12 13 2
3 4 1
4 14 5
4 5 2
5 15 3
5 6 1
6 16 2
6 17 2
7 6 3
```