GRAPH (DFS)

Problem - Keys and Rooms

Medium

https://leetcode.com/problems/keys-and-rooms

```
int maxProfit(vector<int>& prices) {
    int profit = 0;
    int buy = prices[0];
    for (auto i = 1; i < prices.size(); i++) {
        if (prices[i] < buy) {
            buy = prices[i];
        } else if (prices[i] - buy > profit) {
              profit = prices[i] - buy;
        }
    }
    return profit;
}
```

Problem – Clone Graph

Medium

LeetCode https://leetcode.com/problems/clone-graph

Problem Statement

- Given a node reference, create a deep copy of the graph
- The class node has two variables: val and neighbours

```
class Node {
  public int val;
  public List<Node> neighbors;
}
```

- **Output** is the node reference of the copy

Solution – Clone Graph

Medium

LeetCode https://leetcode.com/problems/clone-graph

Solution

- First check the edge cases (is the node null?)
- Create a hash map to store the nodes that is already created

  `unordered<int, Node*> graph;`
- Check if the current node already exists in the graph
- If not, create a new Node object and store in the hashmap
- Visit all the neighbors and add the neighbors to this current node

Code – Clone Graph

Medium

LeetCode https://leetcode.com/problems/clone-graph

```
std::unordered_map<int, Node*> graph;

Node* cloneGraph(Node* node) {
    if (node == NULL) {
        return NULL;
    }
    // does this node object exists?
    if (graph.find(node->val) == graph.end()) {
        // node wasn't visited yet, store in the hashmap
        graph[node->val] = new Node(node->val);
        // visit all neighnours
        for (const auto& n : node->neighbors) {
            graph[node->val]->neighbors.push_back(cloneGraph(n));
        }
    }
    return graph[node->val];
}
```

Problem – 207. Course Schedule

Medium

LeetCode leetcode.com/problems/course-schedule

Problem

- ...

Solution – 207. Course Schedule

Medium

LeetCode leetcode.com/problems/course-schedule

Solution

- ...

Code – 207. Course Schedule

Medium

LeetCode leetcode.com/problems/course-schedule

Code

Time: **O(-)** Space: **O(-)**

- ...

Problem – 417. Pacific Atlantic Water Flow

Medium

LeetCode leetcode.com/problems/pacific-atlantic-water-flow

Problem

- ...

Solution – 417. Pacific Atlantic Water Flow

Medium

LeetCode leetcode.com/problems/pacific-atlantic-water-flow

Solution

- ...

Code – 417. Pacific Atlantic Water Flow

Medium

LeetCode leetcode.com/problems/pacific-atlantic-water-flow

Code

Time: **O(-)** Space: **O(-)**

- ...

Problem – 200. Number of Islands

Medium

LeetCode leetcode.com/problems/number-of-islands

Problem

- ...

Solution – 200. Number of Islands

Medium

LeetCode leetcode.com/problems/number-of-islands

Solution

- ...

Code – 200. Number of Islands

Medium

LeetCode leetcode.com/problems/number-of-islands

Code

Time: **O(-)** Space: **O(-)**

- ...

Problem – 128. Longest Consecutive Sequence

Medium

LeetCode leetcode.com/problems/longest-consecutive-sequence

Problem

- ...

Solution – 128. Longest Consecutive Sequence

Medium

LeetCode leetcode.com/problems/longest-consecutive-sequence

Solution

- ...

Code – 128. Longest Consecutive Sequence

Medium

LeetCode leetcode.com/problems/longest-consecutive-sequence

Code

Time: **O(-)** Space: **O(-)**

- ...

Problem – 261. Graph Valid Tree

Medium

LeetCode leetcode.com/problems/graph-valid-tree

Problem

- ...

Solution – 261. Graph Valid Tree

Medium

LeetCode leetcode.com/problems/graph-valid-tree

Solution

- ...

Code – 261. Graph Valid Tree

Medium

LeetCode leetcode.com/problems/graph-valid-tree

Code

Time: **O(-)** Space: **O(-)**

- ...

Problem – 323. Number of Connected Components

Medium

LeetCode leetcode.com/problems/number-of-connected-components-in-an-undirected-graph

Problem

- ...

Problem – 323. Number of Connected Components

Medium

LeetCode leetcode.com/problems/number-of-connected-components-in-an-undirected-graph

Solution

- ...

Problem – 323. Number of Connected Components

Medium

LeetCode leetcode.com/problems/number-of-connected-components-in-an-undirected-graph

Code

Time: **O(-)** Space: **O(-)**

- ...