

Combination Sum

[2, 3, 6, 7]

T = 7

// action

// Return

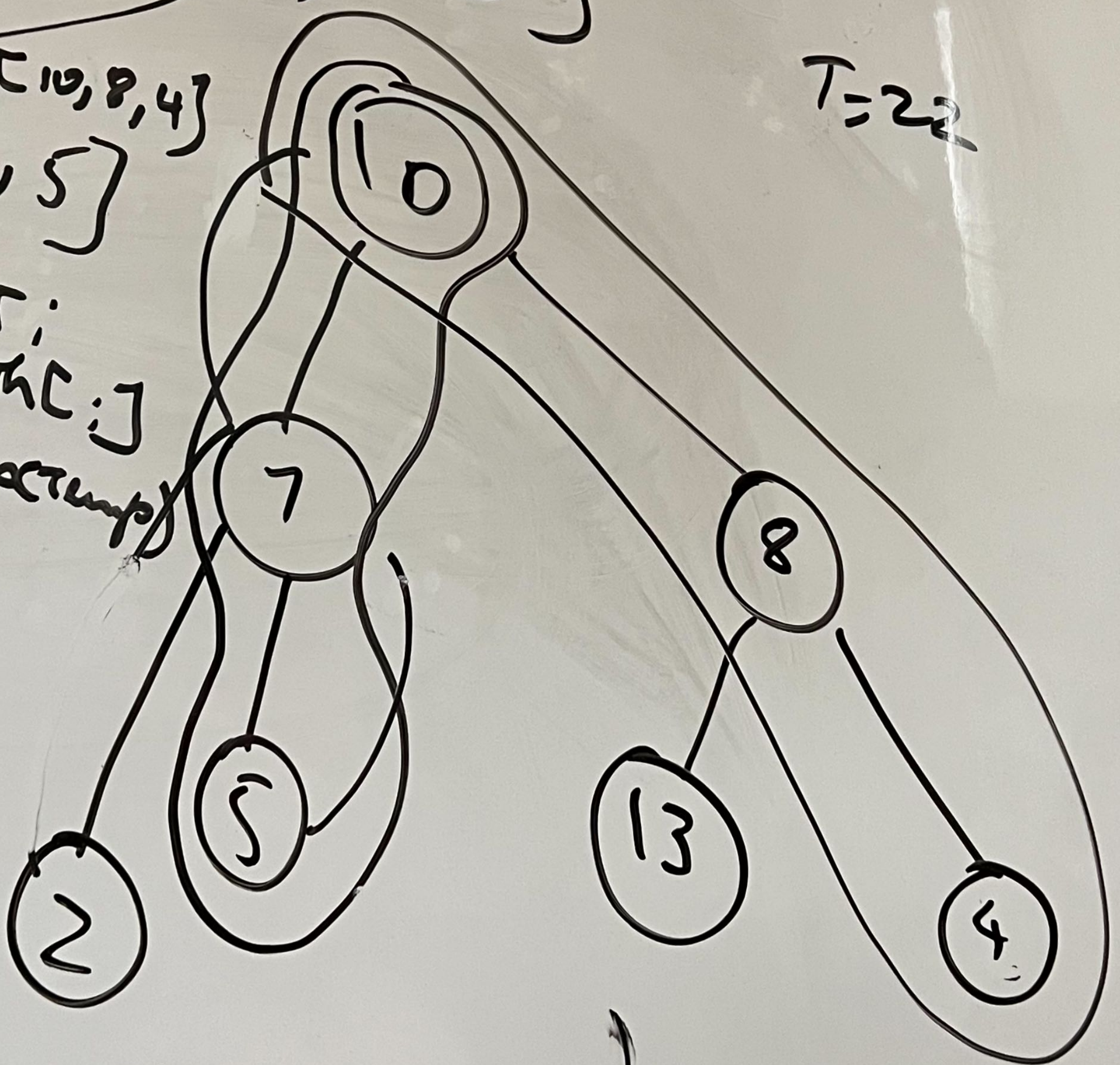
// backtrack
Path*

[10, 7, 8, 8, 1, 4]
[], []

[10, 8, 4]

[10, 7, 5]

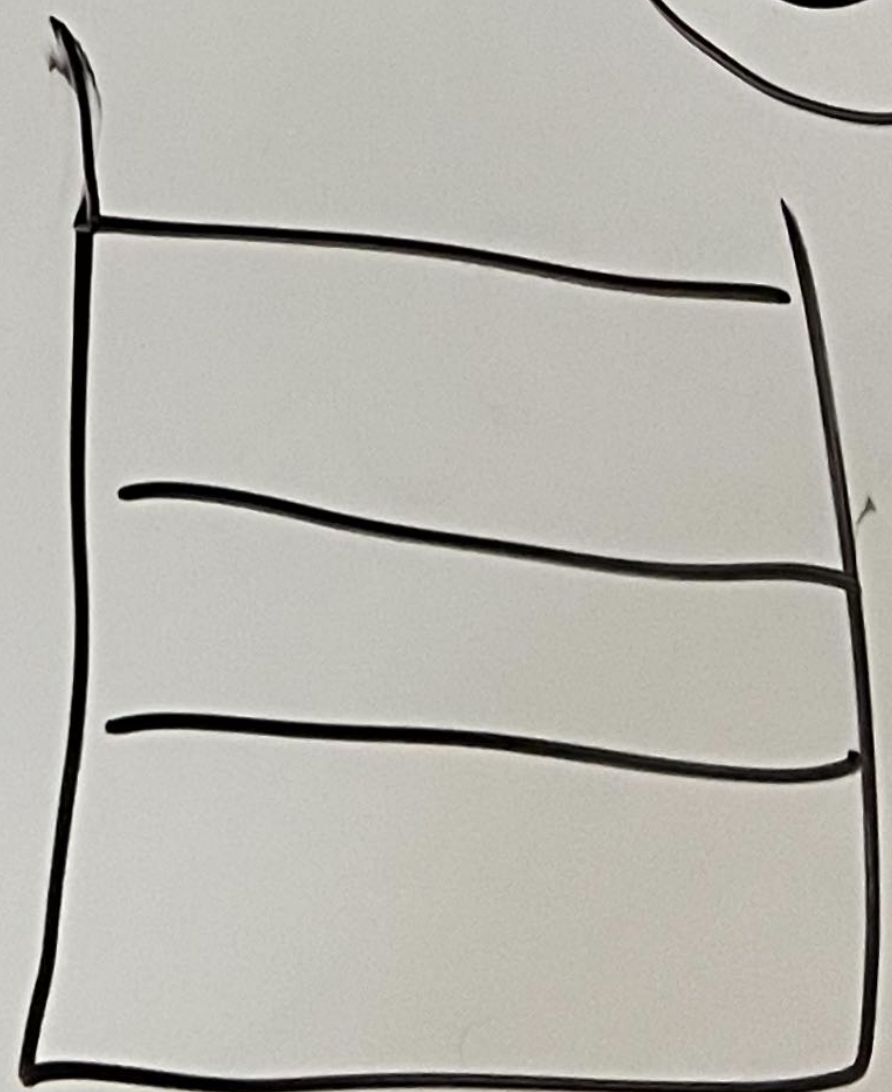
if cur == T:
temp = Path[:]
result.append(temp)



T = 22

// backtrack
pop()

Going back to
Previous state



- Design DS

- Hashmap

- Binary Search

- Array

- Two pointers

- Dynamic Programming

- Heaps

- Monotonic Stack

- Linked list

- Trees

- BFS

- DFS

- Backtracking

- Graph

| | | | | |
|---|--|---|---|---|
| | | 1 | | |
| | | | | 1 |
| | | | | |
| 1 | | | | |
| | | | 1 | |

| | | | | |
|---|--|---|---|---|
| | | | 1 | |
| 1 | | | | |
| | | | | 1 |
| | | | | |
| | | 1 | | |

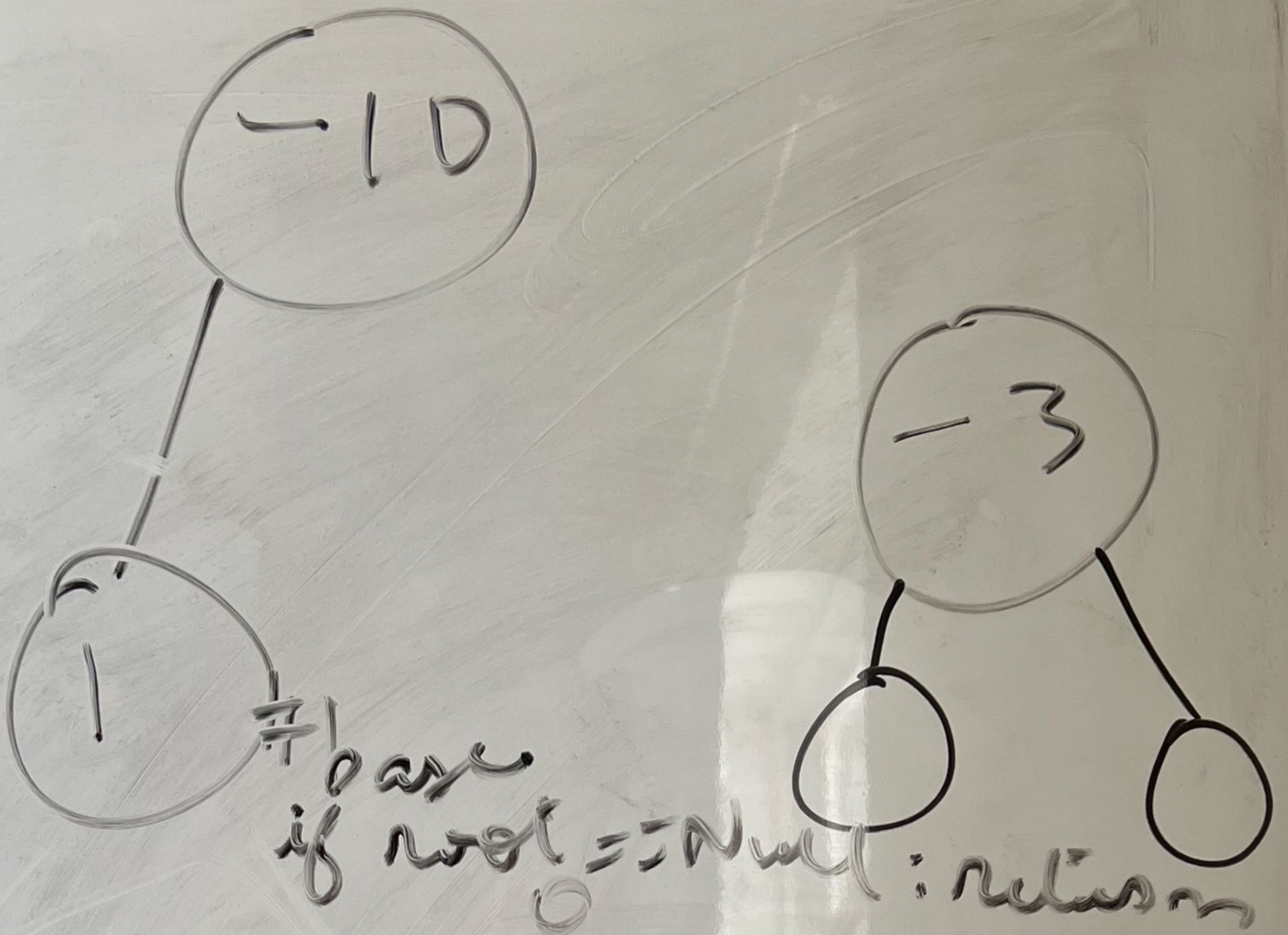
- Tries

- Design (LRU cache, Autocomplete, Logging Rate)

- Sliding Window

- DP2

✓ + 2 ✓
 ✓ ✓



gm = $l + r + n_i = 0 + -3 + 0$

$l_s = 0$ (dfs. left)

$r_s = 0$ (dfs. right)

Return $\text{root} + \text{max}(l_s, r_s)$
 Val

Maximum Path Sum

-2

