DYNAMIC PROGRAMMING 2

DP CATEGORIES

- Matrix type
  - https://github.com/Brady31027/leetcode/blob/master/reports/dynamic_programming.pdf
- Sequence type
  - Single sequence
  - Multiple sequences
- Backpack type

SEQUENCE TYPE DP

- State
  - State at certain node in a sequence
  - Cost, steps, …etc
- Function
  - State transition relationship

NO. 70

CLIMBING STAIRS

PROBLEM DESCRIPTION

- You are climbing a stair case. It takes n steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

- Note: Given n will be a positive integer.

IDEAS

- To reach the N^{th} stair
  - From the $N-1^{th}$ stair
    - How many approaches to reach the $N-1^{th}$ stair?
      - From $N-2^{th}$ stair
      - From $N-3^{th}$ stair
  - From the $N-2^{th}$ stair -- $N-3^{th}$ + $N-4^{th}$
- Ways(n) = Ways(n-1) + Ways(n-2)

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/70_Climbing_Stairs

NO. 198

HOUSE ROBBER

PROBLEM DESCRIPTION

- You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security system connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

- Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight without alerting the police.

IDEAS

- State
  - Possible maximum money
- Initialization
  - state[0] = money[0]
  - state[1] = max(money[0], money[1])
- Function
  - state[n] = max(state[n-2]+money[n], state[n-1])

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/198_House_Robber

NO. 55

JUMP GAME

PROBLEM DESCRIPTION

- Given an array of non-negative integers, you are initially positioned at the first index of the array. Each element in the array represents your maximum jump length at that position. Determine if you are able to reach the last index.
- For example:
- A = [2,3,1,1,4], return true.
- A = [3,2,1,0,4], return false.

IDEAS

- State
  - How high you can jump in the index
- Function
  - state[n] = max(state[n-1]-1, jump[n])
- Init
  - state[0] = jump[0]
- If state is 0, return false

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/55_Jump_Game