**Justin\_Array\_DynamicProgramming\_0746**

**Concept:**

先把到每個點的最小值記錄到一個 list

到每個點的最小值 = min((到前前點的最小值+前前點的值), (到前一點的最小值+前一點的值))

**Code:**

class Solution:

def minCostClimbingStairs(self, cost: List[int]) -> int:

if len(cost) == 0:

return 0

elif len(cost) == 1:

return 0

elif len(cost) == 2:

return min(cost[0], cost[1])

n = len(cost)

dp = [0] \* (n+1)

for i in range(2, n+1):

dp[i] = min(dp[i- 2] + cost[i - 2], dp[i - 1] + cost[i - 1])

return dp[-1]