**Justin\_DynamicProgramming\_0256. Paint House**

**Concept:**

現在的 = 現在的 + 前一個的min(其他兩個)

#要從後面加前面的

current\_row[0] += min(previous\_row[1], previous\_row[2])

current\_row[1] += min(previous\_row[0], previous\_row[2])

current\_row[2] += min(previous\_row[0], previous\_row[1])

**Code:**

class Solution:

def minCost(self, costs: List[List[int]]) -> int:

if len(costs) == 0:

return 0

previous\_row = costs[-1]

for i in reversed(range(len(costs) - 1)):

current\_row = costs[i]

print("prev = ",previous\_row)

print("curr = ",current\_row)

current\_row[0] += min(previous\_row[1], previous\_row[2])

current\_row[1] += min(previous\_row[0], previous\_row[2])

current\_row[2] += min(previous\_row[0], previous\_row[1])

previous\_row = current\_row

return min(previous\_row)