

{% note info %} 摘要 Title: 282. 石子合并 Tag: 区间dp Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

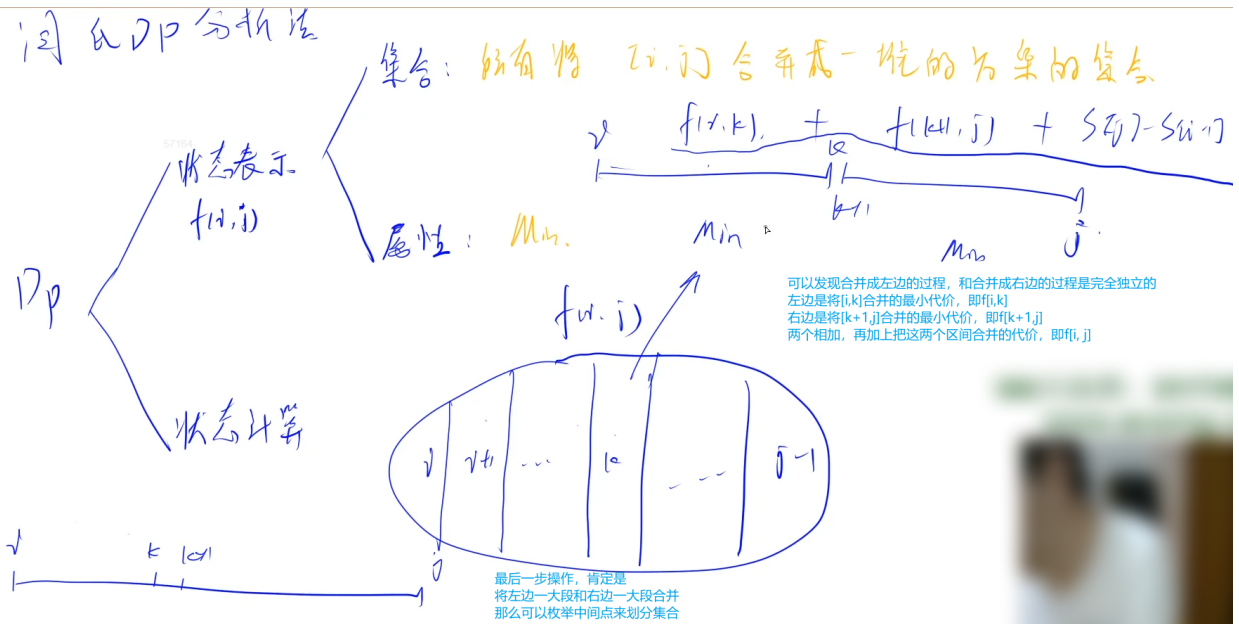
282. 石子合并

• 题意

设有 N 堆石子排成一排，其编号为 $1, 2, 3, \dots, N$ 。每堆石子有一定的质量，可以用一个整数来描述，现在要将这 N 堆石子合并成为一堆。每次只能合并相邻的两堆，合并的代价为这两堆石子的质量之和，合并后与这两堆石子相邻的石子将和新堆相邻，合并时由于选择的顺序不同，合并的总代价也不相同。问题是：找出一种合理的方法，使总的代价最小，输出最小代价。

• 思路

复杂度 $O(n^3)$



• 代码

```
...
Author: NEFU AB-IN
Date: 2022-03-07 20:22:14
FilePath: \ACM\Acwing\282.py
LastEditTime: 2022-03-07 20:30:56
...
N = 310
INF = int(2e9)
```

```
dp = [[0] * N for _ in range(N)]
s = [0] * N

n = int(input())
s[1:] = map(int, input().split())
for i in range(1, n + 1):
    s[i] += s[i - 1]

for len in range(2, n + 1): #先枚举区间长度
    for i in range(1, n - len + 2): #枚举左端点, 边界为 i + len - 1 <= n
        j = i + len - 1
        dp[i][j] = INF
        for k in range(i, j + 1): #再枚举分界点
            dp[i][j] = min(dp[i][j], dp[i][k] + dp[k + 1][j] + s[j] - s[i - 1])

print(dp[1][n])
```