

2551. Put Marbles in Bags

Hard Topics Companies Hint

You have k bags. You are given a 0-indexed integer array `weights` where `weights[i]` is the weight of the i^{th} marble. You are also given the integer k .

Divide the marbles into the k bags according to the following rules:

- No bag is empty.
- If the i^{th} marble and j^{th} marble are in a bag, then all marbles with an index between the i^{th} and j^{th} indices should also be in that same bag.
- If a bag consists of all the marbles with an index from i to j inclusively, then the cost of the bag is `weights[i] + weights[j]`.

The **score** after distributing the marbles is the sum of the costs of all the k bags.

Return the **difference** between the **maximum** and **minimum** scores among marble distributions.

Example 1:

Input: `weights = [1,3,5,1]`, `k = 2`

Output: 4

Explanation:

The distribution `[1],[3,5,1]` results in the minimal score of $(1+1) + (3+1) = 6$.

The distribution `[1,3],[5,1]`, results in the maximal score of $(1+3) + (5+1) = 10$.

Thus, we return their difference $10 - 6 = 4$.

→ must have $k-1$ slices

→ stars & bars

$$\sum_{n=i}^{n=j} \text{marble}[n]$$

$$= \text{marble}[i] + \text{marble}[j]$$

observation:

① $w[0], w[-1]$ are added at least once.

② choose $2(k-1)$ values from w , without duplication.

③ if $w[i]$ is chosen, $w[i+1]$ is a must
↑
apart from head and tail,

→ w_1, w_2, \dots, w_n, k

put $k-1$ bar btw w ,

add up the val of w on the right & left side of the bar, use this to find max and min.

→ $\text{bar_sum}[0] \times (k-1)$

$$\text{bar_sum}[i] = w[i] + w[i+1]$$

`bar_sum.sort()`

`max = largest (k-1) pair`

`min = smallest (k-1) pair`

`return max-min`

Accepted 103 / 103 testcases passed

AndrewC275 submitted at Mar 31, 2025 11:00

Editorial

Solution

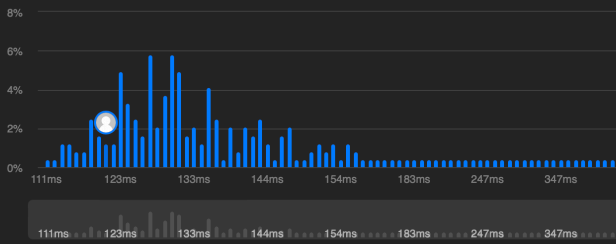
Runtime

121 ms | Beats 90.91%

Analyze Complexity

Memory

30.18 MB | Beats 86.78%



```
1 class Solution:
2     def putMarbles(self, weights: List[int], k: int) -> int:
3         n = len(weights)
4         if n == k or k == 1:
5             return 0
6
7         bar_sum = []
8
9         for i in range(n-1):
10             bar_sum.append(weights[i] + weights[i+1])
11
12         bar_sum.sort()
13
14         mmax = sum(bar_sum[-k+1:])
15         mmin = sum(bar_sum[:k-1])
16
17         return mmax - mmin
18
```