Problem 1388: Pizza With 3n Slices

Problem Information

Difficulty: Hard
Acceptance Rate: 53.56%
Paid Only: No
Tags: Array, Dynamic Programming, Greedy, Heap (Priority Queue)

Problem Description

There is a pizza with `3n` slices of varying size, you and your friends will take slices of pizza as follows:

* You will pick **any** pizza slice. * Your friend Alice will pick the next slice in the anti-clockwise direction of your pick. * Your friend Bob will pick the next slice in the clockwise direction of your pick. * Repeat until there are no more slices of pizzas.

Given an integer array `slices` that represent the sizes of the pizza slices in a clockwise direction, return _the maximum possible sum of slice sizes that you can pick_.

Example 1:

![](https://assets.leetcode.com/uploads/2020/02/18/sample_3_1723.png)

Input: slices = [1,2,3,4,5,6] **Output:** 10 **Explanation:** Pick pizza slice of size 4, Alice and Bob will pick slices with size 3 and 5 respectively. Then Pick slices with size 6, finally Alice and Bob will pick slice of size 2 and 1 respectively. Total = 4 + 6.

Example 2:

![](https://assets.leetcode.com/uploads/2020/02/18/sample_4_1723.png)

Input: slices = [8,9,8,6,1,1] **Output:** 16 **Explanation:** Pick pizza slice of size 8 in each turn. If you pick slice with size 9 your partners will pick slices of size 8.

Constraints:

* `3 * n == slices.length` * `1 <= slices.length <= 500` * `1 <= slices[i] <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int maxSizeSlices(vector<int>& slices) {

}
};
```

Java:

```
class Solution {
public int maxSizeSlices(int[] slices) {

}
}
```

Python3:

```
class Solution:
def maxSizeSlices(self, slices: List[int]) -> int:
```