1167. Minimum Cost to Connect Sticks

Medium 894 145 Add to List Share

You have some number of sticks with positive integer lengths. These lengths are given as an array `sticks`, where `sticks[i]` is the length of the i^{th} stick.

You can connect any two sticks of lengths `x` and `y` into one stick by paying a cost of `x + y`. You must connect all the sticks until there is only one stick remaining.

Return *the minimum cost of connecting all the given sticks into one stick in this way*.

Example 1:

```
Input: sticks = [2,4,3]
Output: 14
Explanation: You start with sticks = [2,4,3].
1. Combine sticks 2 and 3 for a cost of 2 + 3 = 5. Now you have sticks = [5,4].
2. Combine sticks 5 and 4 for a cost of 5 + 4 = 9. Now you have sticks = [9].
There is only one stick left, so you are done. The total cost is 5 + 9 = 14.
```

Example 2:

```
Input: sticks = [1,8,3,5]
Output: 30
Explanation: You start with sticks = [1,8,3,5].
1. Combine sticks 1 and 3 for a cost of 1 + 3 = 4. Now you have sticks = [4,8,5].
2. Combine sticks 4 and 5 for a cost of 4 + 5 = 9. Now you have sticks = [9,8].
3. Combine sticks 9 and 8 for a cost of 9 + 8 = 17. Now you have sticks = [17].
There is only one stick left, so you are done. The total cost is 4 + 9 + 17 = 30.
```

Example 3:

```
public class Solution {
    public int ConnectSticks(int[] sti
        if(sticks.Lengt
        {
            return 0;
        }

        var heap= new H
List<int>(), false);
        foreach(int i i
        {
            heap.Add(i)
        }

        int minCost = 0
        while(heap.Size
        {
            int first =
heap.ExtractTop();
            int next =
heap.ExtractTop();
            int combine
first+next;
            minCost +=
            heap.Add(co
        }
        return minCost;
    }

    public class Heap
        {
            public List
{ get; set; }
            public bool
get; set; }
            public
```

Testcase Run Code Result

Accepted Runtime: 105 ms

Your input [2,4,3]

Output 14

Expected 14

Console Use Example Testcase