

Maximal Score After Applying K Operations

🕒 solved by	Senan
🌐 Platform	LeetCode
🔧 difficulty	Medium
# Serial	2530
≡ tags	Heaps Maths Sorting
🔗 language	C++
📅 solved on	@14/10/2024
🔗 link	https://leetcode.com/problems/maximal-score-after-applying-k-operations/description/
☑ Completion	✓

Intuition

The problem involves maximizing the sum of selected elements from a given list by repeatedly selecting the maximum element, adding it to the sum, and then reducing its value. The optimal way to maximize the sum is by always selecting the current largest element, adding it to the result, and pushing the reduced value (by a factor of 3, rounded up) back into the list. We repeat this process `k` times.

Approach

1. Use a max-heap (priority queue) to always extract the largest element efficiently.
2. In each iteration:
 - Extract the current largest element from the heap.
 - Add the extracted value to the result.
 - Push back the reduced value (i.e., the largest value divided by 3 and rounded up).
3. Repeat this process `k` times.
4. Return the accumulated sum as the result.

Complexity

Time Complexity:

- **Heap creation:** Inserting all elements into the heap takes $O(n \log n)$, where n is the number of elements in the input array.
- **Iterations:** In each of the k iterations, extracting the maximum and inserting the reduced value both take $O(\log n)$. Thus, the loop takes $O(k \log n)$.

Overall time complexity: $O(n \log n + k \log n)$.

Space Complexity:

- **Heap storage:** The priority queue stores at most n elements at any point, so the space complexity is $O(n)$.

Code

```
class Solution {
public:
    long long maxKelements(vector<int>& nums, int k) {
        // Max-heap to always pick the largest element
        priority_queue<int> pq(nums.begin(), nums.end());

        long long maxScore = 0;
        for(int i = 0; i < k; i++) {
            double currScore = pq.top();
            pq.pop();
            maxScore += currScore;
            pq.push(ceil(currScore / 3));
        }
        return maxScore;
    }
};
```