🪨 1046. Last Stone Weight

Difficulty: Easy

Topic: Heap / Priority Queue

Tags: `Greedy`, `Heap`, `Priority Queue`

📜 Problem Description

You are given an array of integers `stones`, where `stones[i]` is the weight of the *i-th* stone.

We are playing a game where in each turn:

- Select the two heaviest stones.
- Smash them together:
  - If both stones have equal weight, both are destroyed.
  - If not, the smaller one is destroyed, and the bigger one becomes `y - x`.

Repeat this until at most one stone is left.

Return the weight of the last stone. If no stone is left, return `0`.

💡 Thought Process / Intuition

🔧 Idea:

- Use a **Max Heap** to always get the 2 largest stones in each round.
- Push all stones into it.
- In every iteration:
  - Pop top 2 stones.
  - If they are not equal, push their difference.
- At the end:

- Return the last stone or 0.

✅ Code

```
class Solution {
public:
    int lastStoneWeight(vector<int>& stones) {
        priority_queue<int>pq;
        for(int ele:stones){
            pq.push(ele);
        }
        while(pq.size()>1){
            int x = pq.top();
            pq.pop();
            int y = pq.top();
            pq.pop();
            if(x!=y) pq.push(x-y);
        }
        if(pq.size()>0){
            return pq.top();
        }
        else return 0;
    }
};
```

🧱 Time and Space Complexity

Complexity Type	Value	Explanation
🕒 Time	O(N log N)	Each insert/pop in heap = log N, repeated N times
🗃️ Space	O(N)	Priority Queue stores all stones

🔄 Dry Run (with 🪨 Emojis)

Input: `stones = [2, 7, 4, 1, 8, 1]`

Step	Max Heap (PQ)	Operation Performed	Remaining Stones
1️⃣	[8, 7, 4, 1, 2, 1]	Pop 8 & 7 → Push 1 (8-7)	[4, 2, 1, 1, 1]
2️⃣	[4, 2, 1, 1, 1]	Pop 4 & 2 → Push 2 (4-2)	[2, 1, 1, 1]
3️⃣	[2, 1, 1, 1]	Pop 2 & 1 → Push 1 (2-1)	[1, 1, 1]
4️⃣	[1, 1, 1]	Pop 1 & 1 → Same → Both destroyed	[1]
✅	[1]	One stone left → Return 1	

🧠 Final Output: `1`

📌 Takeaways

- Max Heap is best for retrieving highest values quickly.
- Greedy logic to simulate stone smashing.
- Very handy for interviews to showcase **Heap usage**.