

1046. Last Stone Weight

07 April 2022 07:54 AM

You are given an array of integers `stones` where `stones[i]` is the weight of the i^{th} stone.

We are playing a game with the stones. On each turn, we choose the **heaviest two stones** and smash them together. Suppose the heaviest two stones have weights x and y with $x \leq y$. The result of this smash is:

- If $x == y$, both stones are destroyed, and
- If $x \neq y$, the stone of weight x is destroyed, and the stone of weight y has new weight $y - x$.

At the end of the game, there is **at most one** stone left.

Return *the smallest possible weight of the left stone*. If there are no stones left, return `0`.

Example 1:

```
Input: stones = [2,7,4,1,8,1]
Output: 1
Explanation:
We combine 7 and 8 to get 1 so the array converts to
[2,4,1,1,1] then,
we combine 2 and 4 to get 2 so the array converts to
[2,1,1,1] then,
we combine 2 and 1 to get 1 so the array converts to
[1,1,1] then,
we combine 1 and 1 to get 0 so the array converts to [1]
then that's the value of the last stone.
```

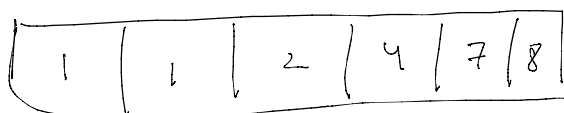
* if $x == y$ both stones are destroyed and

* $x \neq y$, the stone of weight x is destroyed, and the stone of weight y has new weight $y - x$

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

so we first look for heavy stone

for that we use `max_heap`



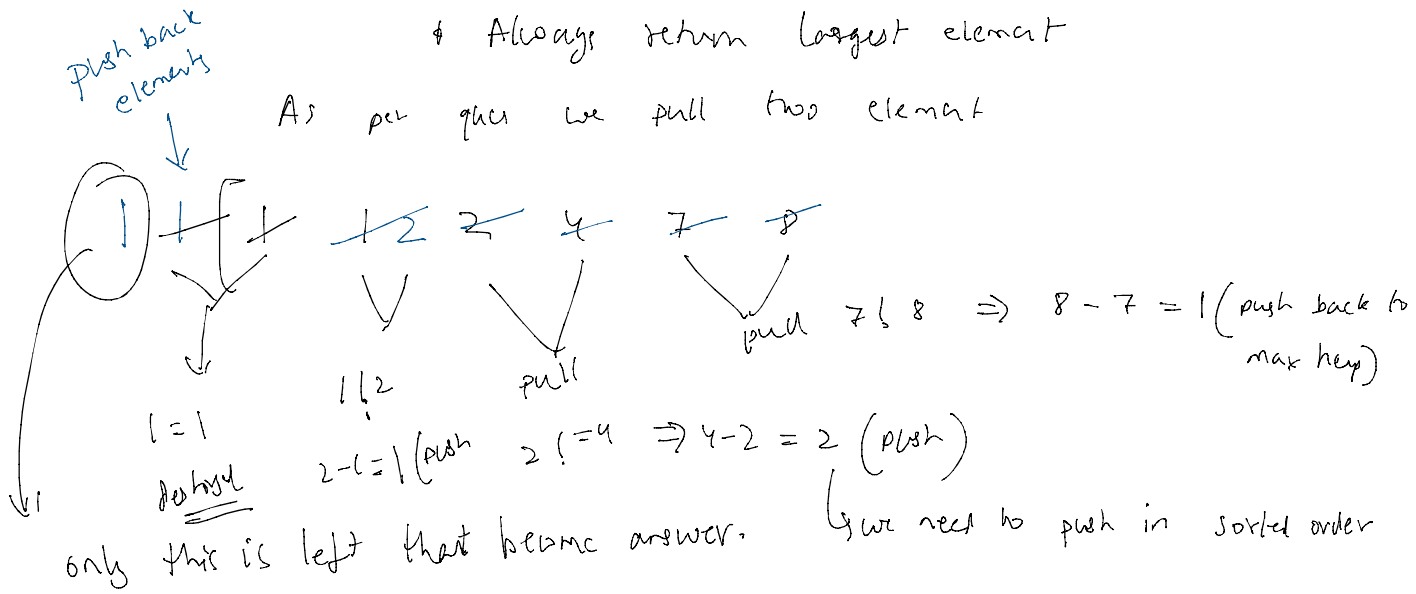
keep

↳ Max-Heap structure is maintained

keep

* Always return largest element

As per given we pull two element



only this is left that become answer. \hookrightarrow we need to push in sorted order

```

public int lastStoneWeight(int[] stones) {
    PriorityQueue<Integer> pq = new PriorityQueue<Integer>((a,b) -> (b-a)); //Max heap (b -a) responsible for it
    for(int stone: stones){
        pq.offer(stone);
    }

    while(pq.size() > 1){
        int largestEl = pq.poll();
        int secondlargestEl = pq.poll();
        // condition given in question
        if(largestEl == secondlargestEl){
            continue;
        } else {
            int diff = largestEl - secondlargestEl;
            pq.offer(diff);
        }
    }
    return pq.size() == 0 ? 0 : pq.poll();
}

```