enqueue 4.

max	min.
~~5~~ 3 < 4	~~4~~ 5
	10
	15
1	20.

enqueue 5.

max	min.
5	~~5~~
3	10.
1.	15.
	20.

```
minHeap.enqueue(num);
maxHeap.enqueue(minHeap.dequeue())
```

↳ to ensure all elements in minHeap is greater than all elements in maxHeap.

```
if(!(minHeap.length >= maxHeap.length)) {
    minHeap.enqueue(maxHeap.dequeue());
}
```

↳ to ensure the length of minHeap is always gte than the length of maxHeap.

Conclusion:

1. maintain two heaps.
   - min heaps → store elements greater than median.
   - max heaps → store elements smaller than median.
2. min heap length is always gte than max heap.

↳ this is because how we do in findMedian.