We use two heaps to satisfy our specification in $O(\log(n))$ time. Our first heap keeps track of our lower-half-valued elements, and the second heap keeps track of the greater-half-valued elements. We maintain an invariant that these two heaps are at most one element apart from each other in size, thus easily allowing us to find the median of our inserted elements.

We maintain a running median, which we set to 0 to begin.

We insert elements one by one, and at each inserted element, we check several cases as follows:

- If our second heap is larger than our first heap:
  - If the inserted element is greater than our current median, we move the top element from the first heap to the second heap, and push the inserted element into the second heap.
  - Otherwise, we push the inserted element into the first heap.
  - We let the median be equal to the top element of the first heap.
- If our first heap is larger than our second heap:
  - If the inserted element is less than the current median, we move the top element from the first heap to the second heap, and push the inserted element into the first heap.
  - Otherwise, we push the inserted element into the second heap.
  - We let the median be equal to the top element of the first heap.
- If both heaps are equal:
  - If the inserted element is less than the current median, we push it into the first heap, and let the median be equal ot the top of the first heap.
  - Otherwise, we push the inserted element into the second heap, and let the median be equal to the top element of the second heap.