

如何构建一个小顶堆?

```
Integer[] data = {  
    70, 30, 34, 73, 60, 68, 43,  
    86, 15, 1, 64, 65, 54, 77,  
    25, 72, 78, 90, 57};  
BinaryHeap<Integer> heap = new BinaryHeap<>(data, new Comparator<Integer>() {  
    public int compare(Integer o1, Integer o2) {  
        return o2 - o1;  
    }  
});
```

Top K问题

- 从 n 个整数中，找出最大的前 k 个数（ k 远远小于 n ）
- 如果使用排序算法进行全排序，需要 $O(n\log n)$ 的时间复杂度
- 如果使用二叉堆来解决，可以使用 $O(n\log k)$ 的时间复杂度来解决
 - 新建一个小顶堆
 - 扫描 n 个整数
 - ✓ 先将遍历到的前 k 个数放入堆中
 - ✓ 从第 $k + 1$ 个数开始，如果大于堆顶元素，就使用 replace 操作（删除堆顶元素，将第 $k + 1$ 个数添加到堆中）
 - 扫描完毕后，堆中剩下的就是最大的前 k 个数
- 如果是找出最小的前 k 个数呢？
 - 用大顶堆
 - 如果小于堆顶元素，就使用 replace 操作

```
for (int i = 0; i < n; i++) {  
    int value = data[i];  
    if (minHeap.size() < k) {  
        minHeap.add(value);  
    } else if (value > minHeap.get()) {  
        minHeap.replace(value);  
    }  
}
```

```
for (int i = 0; i < n; i++) {  
    int value = data[i];  
    if (maxHeap.size() < k) {  
        maxHeap.add(value);  
    } else if (value < maxHeap.get()) {  
        maxHeap.replace(value);  
    }  
}
```