

Minimum increment/decrement to make array non-Increasing

通过+1/-1操作，使得数组称为非递增数组，返回需要的操作次数

小根堆算法，采用 **优先级队列** 实现

首先初始化小根堆初始元素为arr[0]

接着遍历数组，对于当前的元素arr[i]，比较到目前为止的小根堆的堆顶与arr[i]的大小：

1. 如果arr[i]大于堆顶，则说明，需要向这个堆顶元素靠拢才可以满足非递增要求，记录下堆顶元素与arr[i]的差值之后，将堆顶元素弹出，并将arr[i]入堆
2. 如果arr[i]小于堆顶，则直接将arr[i]入堆

这里在步骤1中为什么在比较之后，将堆顶元素弹出？参考下面两个例子

1. arr = {1000, 888, 998, 999}
2. arr = {1, 5, 5, 5}
3. 在arr[i]与堆顶比较之后，只需要保证后续的arr元素都小于等于arr[i]即可，不需要小于等于之前最小的堆顶元素

```
#include <bits/stdc++.h>

using namespace std;

int minCounts(vector<int> &arr) {
    int counts = 0, diff = 0;
    priority_queue<int, vector<int>, greater<>> que;
    for (int a: arr) {
        if (!que.empty() && que.top() < a) {
            diff = a - que.top();
            counts += diff;
            que.pop();
            que.push(a);
        }
        que.push(a);
    }
    return counts;
}
```

```
int main() {  
    vector<int> arr = {3, 1, 2, 1};  
    printf("%d\n", minCounts(arr));  
    return 0;  
}
```