重庆南开中学

单调队列

南开中学信息学竞赛教练组

重庆南开中学

目录

contents

单调队列

- 太简单了

队列

队列

- 符合先进先出原则
- 从队尾进，从队首出
- 可以自己用数组写，也可以用<queue>里的queue来写

类比

- 挤公交
- 所有人都遵守秩序，排队上车
- 先来的人先上车，后来的人后上车
- 没有插队现象，没有中途退出

```
#include <queue>
using namespace std;
int main() {
    queue<double> q;          //创建一个队列，元素是double类型
    q.push(3.14);             //把3.14加入到队尾
    double x = q.front();     //获取队首元素
    q.pop();                  //队首元素出队
    if (q.empty()) {  }       //队列如果为空就执行一些操作
    int s = q.size();         //查询当前队列中的元素个数
    return 0;
}
```

队列

队列

- 符合先进先出原则
- 从队尾进，从队首出
- 可以自己用数组写，也可以用<queue>里的queue来写

```
#include <queue>
using namespace std;
int main() {
    queue<double> q;          //创建一个队列，元素是double类型
    q.push(3.14);             //把3.14加入到队尾
    double x = q.front();     //获取队首元素
    q.pop();                  //队首元素出队
    if (q.empty()) {  }       //队列如果为空就执行一些操作
    int s = q.size();         //查询当前队列中的元素个数
    return 0;
}
```

单调队列

什么叫单调？

- 一些数据从小到大排列——单调递增
- 从大到小排列——单调递减

单调队列

什么叫单调？

- 一些数据从小到大排列——单调递增
- 从大到小排列——单调递减

单调队列

- 单调递增的队列：在队列的基础上，时刻保持队列中的元素单调递增。
- 单调递减的队列：在队列的基础上，时刻保持队列中的元素单调递减。

单调队列

什么叫单调？

- 一些数据从小到大排列——单调递增
- 从大到小排列——单调递减

单调队列

- 单调递增的队列：在队列的基础上，时刻保持队列中的元素单调递增。
- 单调递减的队列：在队列的基础上，时刻保持队列中的元素单调递减。

怎么做到？

- 除了队首可以出队外，新增一个在队尾出队的操作；
- 新元素要进队，先让不满足单调性的元素从队尾出队。

单调队列

什么叫单调？

- 一些数据从小到大排列——单调递增
- 从大到小排列——单调递减

单调队列

- 单调递增的队列：在队列的基础上，时刻保持队列中的元素单调递增。
- 单调递减的队列：在队列的基础上，时刻保持队列中的元素单调递减。

怎么做到？

- 除了队首可以出队外，新增一个在队尾出队的操作；
- 新元素要进队，先让不满足单调性的元素从队尾出队。

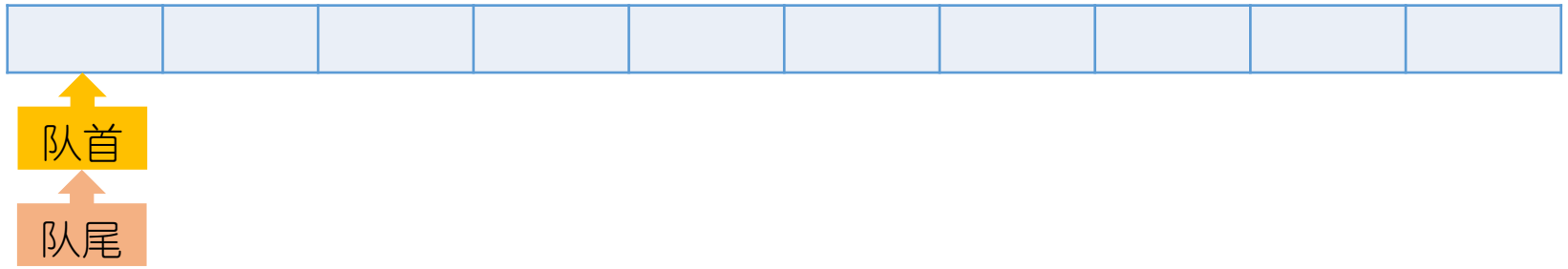
类比

- 挤公交，所有人都很暴躁
- 如果自己比前面的人强壮，就把前面的人打死，除非打不过
- 则排队的人强壮程度总是单调递减

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列



单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队
  - 数字5进队　　不能保持单调性，先将8出队，再让5进队

单调队列

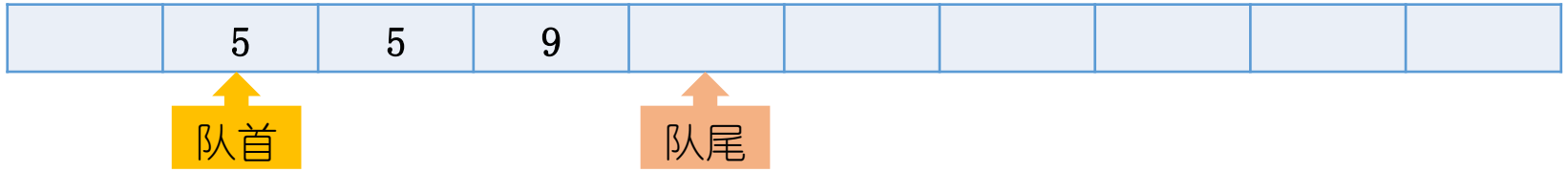
举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队
  - 数字5进队　　不能保持单调性，先将8出队，再让5进队
  - 出队　　数字3出队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队
  - 数字5进队　　不能保持单调性，先将8出队，再让5进队
  - 出队　　数字3出队
  - 数字9进队　　能保持单调性，直接进队



单调队列

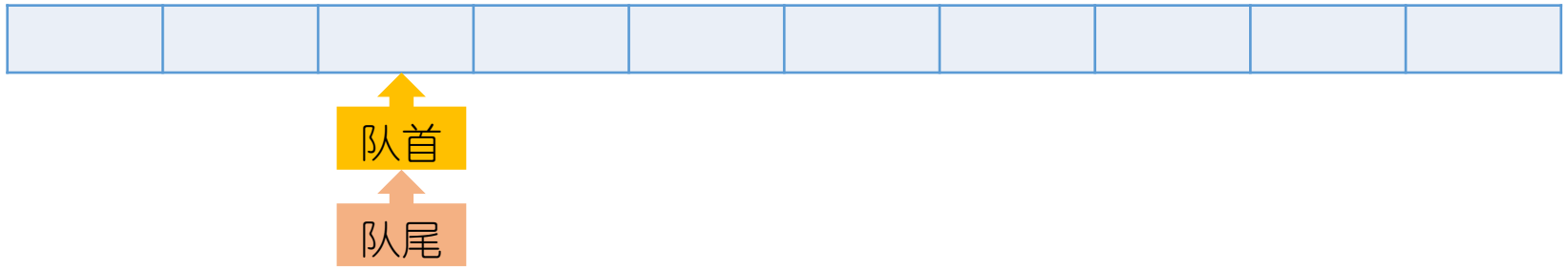
举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队
  - 数字5进队　　不能保持单调性，先将8出队，再让5进队
  - 出队　　数字3出队
  - 数字9进队　　能保持单调性，直接进队
  - 数字4进队　　先让5, 5, 9都出队，再让4进队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队　　空队列，直接进队
  - 数字5进队　　能保持单调性，直接进队
  - 数字8进队　　能保持单调性，直接进队
  - 数字5进队　　不能保持单调性，先将8出队，再让5进队
  - 出队　　数字3出队
  - 数字9进队　　能保持单调性，直接进队
  - 数字4进队　　先让5, 5, 9都出队，再让4进队
  - 出队　　数字4出队



单调队列

怎么写

- 用C++自带的，<deque>中的双端队列deque；
- 头文件可以写#include <queue>，因为头文件<queue>中包含了头文件<deque>。

单调队列

怎么写

- 用C++自带的，<deque>中的双端队列deque；
- 头文件可以写#include <queue>，因为头文件<queue>中包含了头文件<deque>。

```cpp
#include <deque>
using namespace std;
int main() {
    //创建一个队列，元素是double类型，单调递增，允许重复元素
    deque<double> q;

    //向队列中加入元素
    double x = 3.14;
    while (!q.empty() && q.back() > x) {
        q.pop_back();
    }
    q.push_back(x);

    //获取队首元素
    double y = q.front();

    //从队首删除元素
    q.pop_front();

    return 0;
}

```

单调队列

怎么写

- 用C++自带的，<deque>中的双端队列deque；
- 头文件可以写#include <queue>，因为头文件<queue>中包含了头文件<deque>。

时间复杂度

- 每个元素都只会进队1次，出队1次；
- deque内部通过一些技巧让队列两端的长度都可任意调整，且平均每次都是$O(1)$
- 如果有n个元素，总时间复杂度$O(n)$

```cpp
#include <deque>
using namespace std;
int main() {
    //创建一个队列，元素是double类型，单调递增，允许重复元素
    deque<double> q;

    //向队列中加入元素
    double x = 3.14;
    while (!q.empty() && q.back() > x) {
        q.pop_back();
    }
    q.push_back(x);

    //获取队首元素
    double y = q.front();

    //从队首删除元素
    q.pop_front();

    return 0;
}

```

单调队列

怎么写

- 用C++自带的，<deque>中的双端队列deque；
- 头文件可以写#include <queue>，因为头文件<queue>中包含了头文件<deque>。

时间复杂度

- 每个元素都只会进队1次，出队1次；
- deque内部通过一些技巧让队列两端的长度都可任意调整，且平均每次都是$O(1)$
- 如果有n个元素，总时间复杂度$O(n)$

基本性质

- 队列性质：仍在队列中的元素，先进的更靠近队首
- 队列性质：从队首出去的元素，先进先出
- 单调性：队列中的数据有单调性

```cpp
#include <deque>
using namespace std;
int main() {
    //创建一个队列，元素是double类型，单调递增，允许重复元素
    deque<double> q;

    //向队列中加入元素
    double x = 3.14;
    while (!q.empty() && q.back() > x) {
        q.pop_back();
    }
    q.push_back(x);

    //获取队首元素
    double y = q.front();

    //从队首删除元素
    q.pop_front();

    return 0;
}

```

单调队列

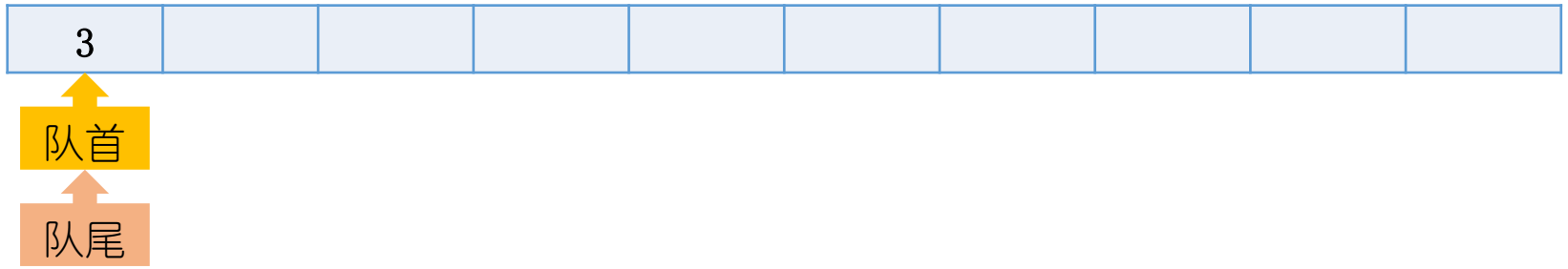
更有用的性质

- 假设是单调递增队列；
- 假设没有队首出队操作，只在队尾进队和出队，即退化为单调栈；
- 性质1：
  - 一个元素进队时，从队尾删除了一些元素
  - 删完后，自己入队之前，“先进队的元素中最后一个比自己小的”恰好在队尾
- 性质2：
  - 一个元素进队时，从队尾删除了一些元素
  - 自己是它们的“后进队的元素中第一个比自己小的”

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列



单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5
  - 数字5进队　　比5先进且≤5的最后一个数是5
    比8后进且＜8的第一个数是5

单调队列

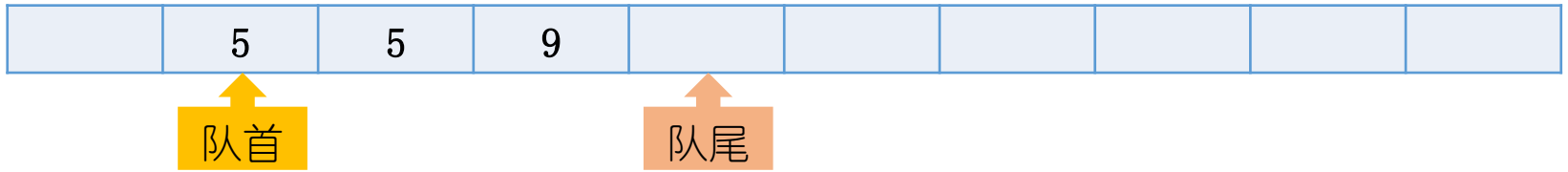
举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5
  - 数字5进队　　比5先进且≤5的最后一个数是5
    比8后进且＜8的第一个数是5
  - 出队　　数字3出队

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5
  - 数字5进队　　比5先进且≤5的最后一个数是5
    比8后进且＜8的第一个数是5
  - 出队　　数字3出队
  - 数字9进队　　比9先进且≤9的最后一个数是5



单调队列

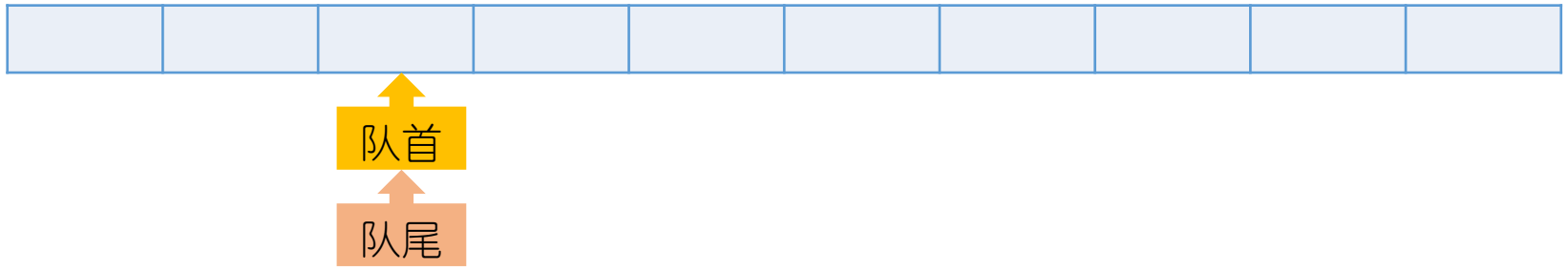
举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5
  - 数字5进队　　比5先进且≤5的最后一个数是5
    比8后进且＜8的第一个数是5
  - 出队　　数字3出队
  - 数字9进队　　比9先进且≤9的最后一个数是5
  - 数字4进队　　比4先进且≤4的最后一个数是3（已出队）
    5, 5, 9每个数，比自己后进且比自己严格小的数都是4

单调队列

举个例子

- 元素是整数，单调递增的队列，允许重复元素
  - 一开始是个空队列
  - 数字3进队
  - 数字5进队　　比5先进且≤5的最后一个数是3
  - 数字8进队　　比8先进且≤8的最后一个数是5
  - 数字5进队　　比5先进且≤5的最后一个数是5
    比8后进且＜8的第一个数是5
  - 出队　　数字3出队
  - 数字9进队　　比9先进且≤9的最后一个数是5
  - 数字4进队　　比4先进且≤4的最后一个数是3（已出队）
    5, 5, 9每个数，比自己后进且比自己严格小的数都是4
  - 出队　　数字4出队



单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

样例输入：
6
3 5 8 5 9 4

样例输出：
-1 1 2 1 4 1

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法1
- 单调递增队列，不允许重复元素；

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法1
- 单调递增队列，不允许重复元素；
- 从左到右枚举每个数，让每个数进队
  - 每个数进队前可能会从队尾删除一些数
  - 删除完成后，队尾的数就是“左边最近的比自己小的数”

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法1：
- 单调递增队列，不允许重复元素；
- 从左到右枚举每个数，让每个数进队
  - 每个数进队前可能会从队尾删除一些数
  - 删除完成后，队尾的数就是“左边最近的比自己小的数”
- 题目要求输出数组下标
  - 可以在队列中存数组下标，删除队尾时比较a[que.back()]与a[i]的大小关系
  - 也可以在队列中存pair或struct。

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法1：
- 单调递增队列，不允许重复元素；
- 从左到右枚举每个数，让每个数进队
  - 每个数进队前可能会从队尾删除一些数
  - 删除完成后，队尾的数就是“左边最近的比自己小的数”
- 题目要求输出数组下标
  - 可以在队列中存数组下标，删除队尾时比较a[que.back()]与a[i]的大小关系
  - 也可以在队列中存pair或struct。

```
scanf("%d", &n);
deque<int> q;
for (int i = 1; i <= n; i ++) {
    scanf("%d", &a[i]);
    while (!q.empty() && a[q.back()] >= a[i]) {      //注意队列中存储的是数组下标
        q.pop_back();
    }
    if (q.empty()) {
        printf("-1\n");
    } else {
        printf("%d\n", q.back());
    }
    q.push_back(i);      //在输出a[i]的答案之后再让a[i]入队
}
```

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法2
- 单调递增队列，允许重复元素

单调队列

例题：

输入一个数组$a_1, a_2, \ldots, a_n$，给每个a_i都找出“左边最近的比自己小的数”的数组下标，如果没有就输出-1。数据范围$n \leq 10^6$

解析：

- 算法2
- 单调递增队列，允许重复元素
- 从右到左枚举每个数，让每个数进队
  - 每个数进队前可能会从队尾删除一些数
  - 从队尾出队时记录答案
  - 最终都没出队的数，答案是-1

单调队列

例题2：【NKOJ2150 广告印刷】

最近，afy决定给TOJ印刷广告，广告牌是刷在城市的建筑物上的，城市里有紧靠着的$N(\leq 400000)$个建筑。afy决定在上面找一块尽可能大的矩形放置广告牌。我们假设每个建筑物都有一个高度，从左到右给出每个建筑物的高度$H_1, H_2, \ldots, H_N$并且我们假设每个建筑物的宽度均为1。要求输出广告牌的最大面积。

样例输入：
6
5 8 4 4 8 4

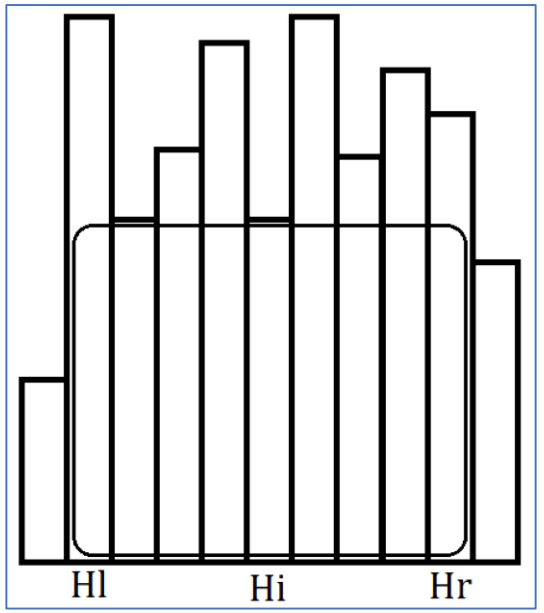
样例输出：
24

单调队列

例题2：【NKOJ2150 广告印刷】

解析：

- 当建筑的范围固定时，广告牌尽量高
  - 底部到地板，顶部受限于最矮的建筑
  - 面积=宽度×范围内最矮的建筑的高度

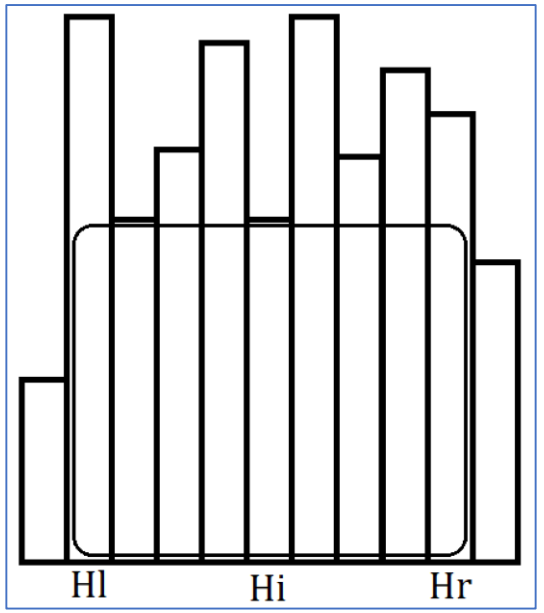


单调队列

例题2：【NKOJ2150 广告印刷】

解析：

- 当建筑的范围固定时，广告牌尽量高
  - 底部到地板，顶部受限于最矮的建筑
  - 面积=宽度×范围内最矮的建筑的高度
- 枚举最矮建筑H_i，广告牌尽量宽
  - 设左边第一个比H_i更矮的建筑是H_{l-1}
  - 设右边第一个比H_i更爱的建筑是H_{r+1}
  - 宽度=$r-l+1$
  - 每个i的l,r可以用单调队列来计算。
- 时间复杂度$O(N)$。



单调队列

例题3：【NKOJ2152 滑动窗口】

输入给你一个长度为$N(\leq 10^6)$的数组，一个长为K的滑动的窗体从最左移至最右端，你只能见到窗口的K个数，每次窗体向右移动一位，找出窗体所包含的数字的最大和最小值。

样例输入：
```
8 3
1 3 -1 -3 5 3 6 7
```

样例输出：
```
-1 -3 -3 -3 3 3
3 3 5 5 6 7
```

样例说明：

[1 3 -1] -3 5 3 6 7	最大=3	最小=-1
1 [3 -1 -3] 5 3 6 7	最大=3	最小=-3
1 3 [-1 -3 5] 3 6 7	最大=5	最小=-3
1 3 -1 [-3 5 3] 6 7	最大=5	最小=-3
1 3 -1 -3 [5 3 6] 7	最大=6	最小=3
1 3 -1 -3 5 [3 6 7]	最大=7	最小=3

单调队列

例题3：【NKOJ2152 滑动窗口】

解析：

- 最大和最小，本质相同，所以先考虑计算最小；

样例输入：
8 3
1 3 -1 -3 5 3 6 7

样例输出：
-1 -3 -3 -3 3 3
3 3 5 5 6 7

单调队列

例题3：【NKOJ2152 滑动窗口】

样例输入：
8 3
1 3 -1 -3 5 3 6 7

样例输出：
-1 -3 -3 -3 3 3
3 3 5 5 6 7

解析：

- 最大和最小，本质相同，所以先考虑计算最小；
- 哪些数没有用？
  - 当$a_3 = -1$进入窗口后，$a_1 = 1, a_2 = 3$都没用了；
  - 当$a_4 = -3$进入窗口后，$a_3 = -1$也没用了；
  - 结论：如果右边有个比自己小的数进入窗口，那自己就没用了。

单调队列

例题3：【NKOJ2152 滑动窗口】

样例输入：
8 3
1 3 -1 -3 5 3 6 7

样例输出：
-1 -3 -3 -3 3 3
3 3 5 5 6 7

解析：

- 最大和最小，本质相同，所以先考虑计算最小；
- 哪些数没有用？
  - 当$a_3 = -1$进入窗口后，$a_1 = 1, a_2 = 3$都没用了；
  - 当$a_4 = -3$进入窗口后，$a_3 = -1$也没用了；
  - 结论：如果右边有个比自己小的数进入窗口，那自己就没用了。
- 单调递增队列，不允许重复元素
  - 模拟滑动窗口，一步步滑动；
  - 删除队尾比a_i大的数，它们没有用了，然后将a_i加入队尾；
  - 如果队首是a_{i-K}，它已经不再窗口内，也删除；
  - 队首元素就是当前窗口的最小值。

单调队列

例题3：【NKOJ2152 滑动窗口】

样例输入：
8 3
1 3 -1 -3 5 3 6 7

样例输出：
-1 -3 -3 -3 3 3
3 3 5 5 6 7

解析：

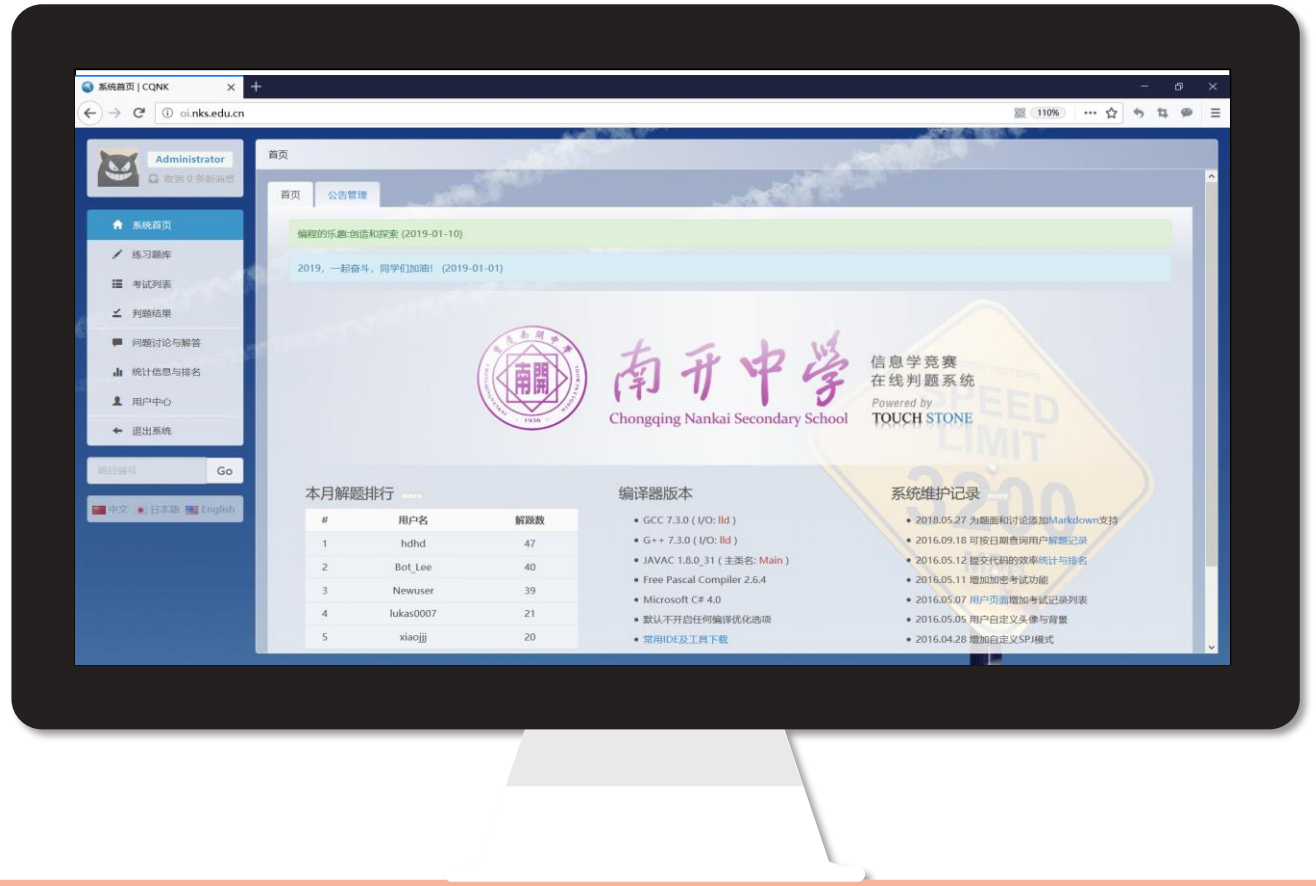
- 最大和最小，本质相同，所以先考虑计算最小；
- 哪些数没有用？
  - 当$a_3 = -1$进入窗口后，$a_1 = 1, a_2 = 3$都没用了；
  - 当$a_4 = -3$进入窗口后，$a_3 = -1$也没用了；
  - 结论：如果右边有个比自己小的数进入窗口，那自己就没用了。
- 单调递增队列，不允许重复元素
  - 模拟滑动窗口，一步步滑动；
  - 删除队尾比a_i大的数，它们没有用了，然后将a_i加入队尾；
  - 如果队首是a_{i-K}，它已经不再窗口内，也删除；
  - 队首元素就是当前窗口的最小值。
- 最大同理，时间复杂度$O(N)$。

课后练习

单调队列习题

NKOJ2150	广告印刷
NKOJ2152	滑动窗口
NKOJ2149	布丁
NKOJ2419	集合的第N大元素
NKOJ1134	合并果子
NKOJ3891	地平线上的城市
NKOJ3955	最大子段和
NKOJ5558	木板倒水
NKOJ5559	静音问题

重庆南开中学


系统首页 | CQNK
oi.nks.edu.cn
Administrator
首页
公告管理
系统首页
练习题库
考试列表
判题结果
问题讨论与解答
统计信息与排名
用户中心
退出系统
Go
编程的乐趣:创造和探索 (2019-01-10)
2019，一起奋斗，同学们加油！ (2019-01-01)
南开中学
Chongqing Nankai Secondary School
信息学竞赛
在线判题系统
Powered by
TOUCH STONE
本月解题排行
编译器版本
系统维护记录


Noi


oi.nks.edu.cn