


常用代码模板1——基础算法

作者:  YXC (<https://www.acwing.com/user/myspace/index/1/>), 2019-07-31 21:18:31, 阅读 25854

瀛

算法基础课相关代码模板

275

- 活动链接 —— 算法基础课 (<https://www.acwing.com/activity/content/11/>)

瀛

快速排序算法模板 —— 模板题 AcWing 785. 快速排序 (<https://www.acwing.com/problem/content/787/>)

→

332

```
void quick_sort(int q[], int l, int r)
{
    if (l >= r) return;

    int i = l - 1, j = r + 1, x = q[l + r >> 1];
    while (i < j)
    {
        do i++; while (q[i] < x);
        do j--; while (q[j] > x);
        if (i < j) swap(q[i], q[j]);
    }
    quick_sort(q, l, j), quick_sort(q, j + 1, r);
}
```

归并排序算法模板 —— 模板题 AcWing 787. 归并排序 (<https://www.acwing.com/problem/content/789/>)

```
void merge_sort(int q[], int l, int r)
{
    if (l >= r) return;

    int mid = l + r >> 1;
    merge_sort(q, l, mid);
    merge_sort(q, mid + 1, r);

    int k = 0, i = l, j = mid + 1;
    while (i <= mid && j <= r)
        if (q[i] < q[j]) tmp[k++] = q[i++];
        else tmp[k++] = q[j++];

    while (i <= mid) tmp[k++] = q[i++];
    while (j <= r) tmp[k++] = q[j++];

    for (i = l, j = 0; i <= r; i++, j++) q[i] = tmp[j];
}
```

整数二分算法模板 —— 模板题 AcWing 789. 数的范围 (<https://www.acwing.com/problem/content/791/>)

```

bool check(int x) { /* ... */ } // 检查x是否满足某种性质

// 区间[l, r]被划分成[l, mid]和[mid + 1, r]时使用：
int bsearch_1(int l, int r)
{
    while (l < r)
    {
        int mid = l + r >> 1;
        if (check(mid)) r = mid; // check()判断mid是否满足性质
        else l = mid + 1;
    }
    return l;
}

// 区间[l, r]被划分成[l, mid - 1]和[mid, r]时使用：
int bsearch_2(int l, int r)
{
    while (l < r)
    {
        int mid = l + r + 1 >> 1;
        if (check(mid)) l = mid;
        else r = mid - 1;
    }
    return l;
}

```

浮点数二分算法模板 —— 模板题 AcWing 790. 数的三次方根 (<https://www.acwing.com/problem/content/792/>)

```

bool check(double x) { /* ... */ } // 检查x是否满足某种性质

double bsearch_3(double l, double r)
{
    const double eps = 1e-6; // eps 表示精度，取决于题目对精度的要求
    while (r - l > eps)
    {
        double mid = (l + r) / 2;
        if (check(mid)) r = mid;
        else l = mid;
    }
    return l;
}

```

高精度加法 —— 模板题 AcWing 791. 高精度加法 (<https://www.acwing.com/problem/content/793/>)

```

// C = A + B, A >= 0, B >= 0
vector<int> add(vector<int> &A, vector<int> &B)
{
    if (A.size() < B.size()) return add(B, A);

    vector<int> C;
    int t = 0;
    for (int i = 0; i < A.size(); i++)
    {
        t += A[i];
        if (i < B.size()) t += B[i];
        C.push_back(t % 10);
        t /= 10;
    }

    if (t) C.push_back(t);
    return C;
}

```

高精度减法 —— 模板题 AcWing 792. 高精度减法 (<https://www.acwing.com/problem/content/794/>)

```
// C = A - B, 满足A >= B, A >= 0, B >= 0
vector<int> sub(vector<int> &A, vector<int> &B)
{
    vector<int> C;
    for (int i = 0, t = 0; i < A.size(); i++)
    {
        t = A[i] - t;
        if (i < B.size()) t -= B[i];
        C.push_back((t + 10) % 10);
        if (t < 0) t = 1;
        else t = 0;
    }

    while (C.size() > 1 && C.back() == 0) C.pop_back();
    return C;
}
```

高精度乘低精度 —— 模板题 AcWing 793. 高精度乘法 (<https://www.acwing.com/problem/content/795/>)

```
// C = A * b, A >= 0, b > 0
vector<int> mul(vector<int> &A, int b)
{
    vector<int> C;

    int t = 0;
    for (int i = 0; i < A.size() || t; i++)
    {
        if (i < A.size()) t += A[i] * b;
        C.push_back(t % 10);
        t /= 10;
    }

    while (C.size() > 1 && C.back() == 0) C.pop_back();

    return C;
}
```

高精度除以低精度 —— 模板题 AcWing 794. 高精度除法 (<https://www.acwing.com/problem/content/796/>)

```
// A / b = C ... r, A >= 0, b > 0
vector<int> div(vector<int> &A, int b, int &r)
{
    vector<int> C;
    r = 0;
    for (int i = A.size() - 1; i >= 0; i--)
    {
        r = r * 10 + A[i];
        C.push_back(r / b);
        r %= b;
    }
    reverse(C.begin(), C.end());
    while (C.size() > 1 && C.back() == 0) C.pop_back();
    return C;
}
```

一维前缀和 —— 模板题 AcWing 795. 前缀和 (<https://www.acwing.com/problem/content/797/>)

$$S[i] = a[1] + a[2] + \dots + a[i]$$

$$a[l] + \dots + a[r] = S[r] - S[l - 1]$$

二维前缀和 —— 模板题 AcWing 796. 子矩阵的和 (<https://www.acwing.com/problem/content/798/>)

$S[i, j]$ = 第*i*行*j*列格子左上部分所有元素的和
以(x_1, y_1)为左上角, (x_2, y_2)为右下角的子矩阵的和为:
 $S[x_2, y_2] - S[x_1 - 1, y_2] - S[x_2, y_1 - 1] + S[x_1 - 1, y_1 - 1]$

一维差分 —— 模板题 AcWing 797. 差分 (<https://www.acwing.com/problem/content/799/>)

给区间 $[l, r]$ 中的每个数加上 c : $B[l] += c, B[r + 1] -= c$

二维差分 —— 模板题 AcWing 798. 差分矩阵 (<https://www.acwing.com/problem/content/800/>)

给以(x_1, y_1)为左上角, (x_2, y_2)为右下角的子矩阵中的所有元素加上 c :
 $S[x_1, y_1] += c, S[x_2 + 1, y_1] -= c, S[x_1, y_2 + 1] -= c, S[x_2 + 1, y_2 + 1] += c$

位运算 —— 模板题 AcWing 801. 二进制中1的个数 (<https://www.acwing.com/problem/content/803/>)

求*n*的第*k*位数字: $n \gg k \& 1$
返回*n*的最后一位1: $\text{lowbit}(n) = n \& -n$

双指针算法 —— 模板题 AcWing 799. 最长连续不重复子序列 (<https://www.acwing.com/problem/content/801/>), AcWing 800. 数组元素的目标和 (<https://www.acwing.com/problem/content/802/>)

```
for (int i = 0, j = 0; i < n; i++)  
{  
    while (j < i && check(i, j)) j++;  
  
    // 具体问题的逻辑  
}
```

常见问题分类:

- (1) 对于一个序列, 用两个指针维护一段区间
- (2) 对于两个序列, 维护某种次序, 比如归并排序中合并两个有序序列的操作

离散化 —— 模板题 AcWing 802. 区间和 (<https://www.acwing.com/problem/content/804/>)

```
vector<int> alls; // 存储所有待离散化的值  
sort(alls.begin(), alls.end()); // 将所有值排序  
alls.erase(unique(alls.begin(), alls.end()), alls.end()); // 去掉重复元素
```

```
// 二分求出x对应的离散化的值  
int find(int x) // 找到第一个大于等于x的位置  
{  
    int l = 0, r = alls.size() - 1;  
    while (l < r)  
    {  
        int mid = l + r >> 1;  
        if (alls[mid] >= x) r = mid;  
        else l = mid + 1;  
    }  
    return r + 1; // 映射到1, 2, ...n  
}
```

区间合并 —— 模板题 AcWing 803. 区间合并 (<https://www.acwing.com/problem/content/805/>)

```
// 将所有存在交集的区间合并
void merge(vector<PII> &segs)
{
    vector<PII> res;

    sort(segs.begin(), segs.end());

    int st = -2e9, ed = -2e9;
    for (auto seg : segs)
        if (ed < seg.first)
        {
            if (st != -2e9) res.push_back({st, ed});
            st = seg.first, ed = seg.second;
        }
        else ed = max(ed, seg.second);


    if (st != -2e9) res.push_back({st, ed});


    segs = res;
}
```


评论列表：


在这里写评论...（支持MarkDown和Latex语法）


提交评论

 bpink (<https://www.acwing.com/user/myspace/index/33955/>) 19天前 回复
(<https://www.acwing.com/user/myspace/index/33955/>)
请问一下，二分查找的两个模板中，这两个模版是应用于不同类型的题目划分的区间，还是只是对于同一种题目的两种解法呢


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 15天前 回复
(<https://www.acwing.com/user/myspace/index/1/>)
一般每道题目这两个模板都可以用，不过会有一个模板更好写一点。


 bpink (<https://www.acwing.com/user/myspace/index/33955/>) 9天前 回复了 yxc 的评论 回
(<https://www.acwing.com/user/myspace/index/33955/>)
get! 谢谢y总~

 jerryflymi (<https://www.acwing.com/user/myspace/index/36946/>) 21天前 回复
(<https://www.acwing.com/user/myspace/index/36946/>)
y总给的快排的模版只是将小于等于x的数放在了左边，大于等于x的数放在了右边，但是并不能实现每一趟都把x放在它最终的位置上。

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 15天前 回复
(<https://www.acwing.com/user/myspace/index/1/>)
对的。


 yanxie411 (<https://www.acwing.com/user/myspace/index/37386/>) 1个月前 回复
(<https://www.acwing.com/user/myspace/index/37386/>)
y总给的快排的模版也适用于数组中有重复元素的情况吗，我试了一下，好像如果有重复元素的话就不行了。能帮忙解答一下吗？


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 1个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>)
适合的，你应该是在某些地方写错了。

 Arroganceの浓 (<https://www.acwing.com/user/myspace/index/29416/>) 2个月前 回复
(<https://www.acwing.com/user/myspace/index/29416/>)
我一直用的二分模版是这样的，区别大吗，我感觉我好像包括了两个模版把


```
int bsearch(int l,int r)
{
    while(l<=r)
    {
        int mid = l+r >> 1;
        if(chekc(mid))r=mid-1;
        else l = mid + 1;
```

```
}  
return l - 1;  
}
```


 Arroganceの浓 (<https://www.acwing.com/user/myspace/index/29416/>) 2个月前 回复
(<https://www.acwing.com/user/myspace/index/29416/>)

 Ripple-zjw (<https://www.acwing.com/user/myspace/index/36526/>) 1个月前 回复了
(<https://www.acwing.com/user/myspace/index/36526/>)
用你键盘左上角的，打六下，把代码放在前三个和后三个之间。然后前三个后面还可以写上你用的是什么语言，比如 cpp`。具体你可以看Markdown语法。


```
int bsearch(int l,int r)  
{  
    while(l<=r)  
    {  
        int mid = l+r >>1;  
        if(chekc(mid))r=mid-1;  
        else l = mid + 1;  
    }  
    return l - 1;  
}
```


 Ripple-zjw (<https://www.acwing.com/user/myspace/index/36526/>) 1个月前 回复了
(<https://www.acwing.com/user/myspace/index/36526/>)


。。。那个点没显示出来，就是这个键~

 Arroganceの浓 (<https://www.acwing.com/user/myspace/index/29416/>) 1个月前 回复了
(<https://www.acwing.com/user/myspace/index/29416/>)


这样吗


 Arroganceの浓 (<https://www.acwing.com/user/myspace/index/29416/>) 1个月前 回复了
(<https://www.acwing.com/user/myspace/index/29416/>)
会了，多谢

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 1个月前 回复了 Arroganceの浓 的评论
(<https://www.acwing.com/user/myspace/index/1/>)
二分模板有很多，背一个经过验证的即可。


 熠丶 (<https://www.acwing.com/user/myspace/index/13827/>) 2个月前 回复


(<https://www.acwing.com/user/myspace/index/13827/>)
在高精度乘法的模板中
vector[HTML_REMOVED] mul(vector[HTML_REMOVED] &A,int B)
为什么要加&?
我去掉&时也能ac
加上&有什么好处吗

 vsbf (<https://www.acwing.com/user/myspace/index/9535/>) 2个月前 回复
(<https://www.acwing.com/user/myspace/index/9535/>)


 minux (<https://www.acwing.com/user/myspace/index/34840/>) 2个月前 回复
(<https://www.acwing.com/user/myspace/index/34840/>)

 熠丶 (<https://www.acwing.com/user/myspace/index/13827/>) 2个月前 回复了 minux 的评论
(<https://www.acwing.com/user/myspace/index/13827/>)
是不是时间换空间？我用两种方法的话看测试数据加&时间会变长

 minux (<https://www.acwing.com/user/myspace/index/34840/>) 2个月前 回复了 熠丶 的评论
(<https://www.acwing.com/user/myspace/index/34840/>)
&仅仅是内存标签作为参数传过去和传地址类似，但是值参数传递就是开辟一块空间将原来参数信息复制到这块空间中，效率上&会高一些

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 2个月前 回复了 熠丶 的评论 回复
(<https://www.acwing.com/user/myspace/index/1/>)

楼上的同学解释地不错，不是时间换空间，而是时间和空间都变小了。


 lxxs (<https://www.acwing.com/user/myspace/index/5870/>) 4个月前 回复


(<https://www.acwing.com/user/myspace/index/5870/>)
用数组做离散化


```


#include<bits/stdc++.h>
#define N 300010
#define For(i,l,r) for(int i = l; i<=r; i++)
using namespace std;
int all[N],allsi;int q[N];
int n,m;
struct node{
    int fir;
    int sec;
};
node add[N],que[N];int addsi,quesi;
int main(){
    cin>>n>>m;
    For(i,1,n){//存储操作并把点保存。
        int x,c;
        cin>>x>>c;
        add[++addsi].fir=x;add[addsi].sec=c;
        all[++allsi]=x;
    }
    For(i,1,m){//存储查询的点
        int l,r;
        cin>>l>>r;
        que[++quesi].fir=l;que[quesi].sec=r;
        all[++allsi]=l;all[++allsi]=r;
    }
    sort(all+1,all+allsi+1);//排序
    allsi=unique(all+1,all+allsi+1)-(all+1);//去重并获取最后一个数的下标
    For(i,1,n){//进行操作。【对离散化后的数组】
        int x = lower_bound(all+1,all+allsi+1,add[i].fir)-all;
        q[x]+=add[i].sec;
    }
    For(i,1,allsi){//计算前缀和
        q[i]+=q[i-1];
    }
    For(i,1,m){//查找离散化后的位置，输出答案
        int l,r;
        l = lower_bound(all+1,all+allsi+1,que[i].fir)-all;
        r = lower_bound(all+1,all+allsi+1,que[i].sec)-all;
        cout<<q[r]-q[l-1]<<endl;
    }
}


```

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 4个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>)


 lxxs (<https://www.acwing.com/user/myspace/index/5870/>) 4个月前 回复了 yxc 的评论 回复
(<https://www.acwing.com/user/myspace/index/5870/>)


 Ker (<https://www.acwing.com/user/myspace/index/28372/>) 4个月前 回复
可否请教一下高精x高精
(<https://www.acwing.com/user/myspace/index/28372/>)


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 4个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>) 比较快的方式是用快速傅里叶变换，可以百度一下。

 Ker (<https://www.acwing.com/user/myspace/index/28372/>) 4个月前 回复了 yxc 的评论 回
(<https://www.acwing.com/user/myspace/index/28372/>)
好的，谢谢大佬

 小白菜_ruc (<https://www.acwing.com/user/myspace/index/15788/>) 6个月前 回复
闫总你好，请问算法基础课程的例题讲解是用cpp来讲吗？谢谢啦！
(<https://www.acwing.com/user/myspace/index/15788/>)

 只想AC一次 (<https://www.acwing.com/user/myspace/index/15362/>) 6个月前 回复
(<https://www.acwing.com/user/myspace/index/15362/>)


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 6个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>)


 发光二极管 (<https://www.acwing.com/user/myspace/index/13176/>) 6个月前 回复
y总感觉mergeSort可以一个while循环搞定y(^ _ ^)V
(<https://www.acwing.com/user/myspace/index/13176/>)

```
void merge_sort(int nums[], int left, int right)
{
    if (left >= right) return ;
    int mid = left + ((right - left) >> 1);
    merge_sort(nums, left, mid);
    merge_sort(nums, mid + 1, right);

    int k = left, l = left, r = mid + 1;
    while (k <= right)
    {
        if ((r > right) || (l <= mid && nums[l] <= nums[r])) tmp[k++] = nums[l++];
        else tmp[k++] = nums[r++];
    }


    for (int i = left; i <= right; ++i) nums[i] = tmp[i];
}
```


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 6个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>) 只要代码正确即可，写法不唯一，选择一种自己最习惯的就好。

 发光二极管 (<https://www.acwing.com/user/myspace/index/13176/>) 6个月前 回复了 yxc 的评论
(<https://www.acwing.com/user/myspace/index/13176/>)
^_^ 嗯嗯




hbhdhd (<https://www.acwing.com/user/myspace/index/10118/>) 7个月前 回复
(<https://www.acwing.com/user/myspace/index/10118/>) 为啥整数二分里 区间[l, r]被划分成[l, mid-1]和[mid, r]时 mid = l+r+1>>1啊


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 7个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>) 否则会有边界问题，当user/myspace/index等于l，那么此时如果执行l=mid 就死循环了。


 hbhdhd (<https://www.acwing.com/user/myspace/index/10118/>) 7个月前 回复了 yxc 的评论
(<https://www.acwing.com/user/myspace/index/10118/>)
晓得了，谢谢大佬




WenQ (<https://www.acwing.com/user/myspace/index/13715/>) 7个月前 回复
(<https://www.acwing.com/user/myspace/index/13715/>) 请问为什么在快排的子函数里， $x = q[l + r >> 1]$ ，时间正常；当我写成 $x = q[l]$ 的时候，会报超时错误呢，他俩在运行时间上有什么差别吗？

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 7个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>) 后来加强了数据，具体可以参考这篇讨论帖及其评论内容：AcWing 785. 注意本题数据已加强
(<https://www.acwing.com/problem/content/discussion/content/395/>)。


 WenQ (<https://www.acwing.com/user/myspace/index/13715/>) 7个月前 回复了 yxc 的评论
(<https://www.acwing.com/user/myspace/index/13715/>)
明白了，谢谢


 minux (<https://www.acwing.com/user/myspace/index/34840/>) 2个月前 回复
(<https://www.acwing.com/user/myspace/index/34840/>) 如果被排序数据具有单调性的话，pivot(选择使用401)的方法将导致时间复杂度从 $O(N \lg N)$ 退化到 $O(N^2)$ ，一般pivot选择选择用mid或者随机化方式降低退化的概率


 yxc (<https://www.acwing.com/user/myspace/index/1/>) 2个月前 回复了 minux 的评论 回复
(<https://www.acwing.com/user/myspace/index/1/>) 对的





Leonardo (<https://www.acwing.com/user/myspace/index/11157/>) 8个月前 回复
(<https://www.acwing.com/user/myspace/index/11157/>) 浮点数二分法是不是-1到1没法算呀？

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 8个月前 回复
(<https://www.acwing.com/user/myspace/index/1/>) 可以算的，终止条件是区间长度和区间两个端点的具体取值无关。


 Leonardo (<https://www.acwing.com/user/myspace/index/11157/>) 8个月前 回复了 yxc 的评论
(<https://www.acwing.com/user/myspace/index/11157/>)
例如：求0.125的三次方根，结果应该是0.5。通过二分，mid越来越小，绝对值小于1的数乘方越来越小。

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 8个月前 回复了 Leonardo 的评论 回
(<https://www.acwing.com/user/myspace/index/1/>)
如果 mid 的三次方小于0.5，就应该选择 [mid, r] 这个区间，否则选择 [l, mid] 这个区间，只要0.5在 [l, r] 中，就一定可以二分出来。

 Leonardo (<https://www.acwing.com/user/myspace/index/11157/>) 8个月前 回复了 yxc 的评论
(<https://www.acwing.com/user/myspace/index/11157/>)
按照模板，求0.125的三次方根时，l=0, r=0.125，是不是mid无法取到大于0.125

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 8个月前 回复了 Leonardo 的评论 回复 (<https://www.acwing.com/user/myspace/index/1/>)


模板里没有说让 $r = 0.125$ 啊, 需要自己根据题目来判断 l 和 r 的取值, 要保证答案一定在 $[l, r]$ 中。

 Leonardo (<https://www.acwing.com/user/myspace/index/11157/>) 8个月前 回复了 yxc 的评论 回复 (<https://www.acwing.com/user/myspace/index/11157/>)

现在明白了, 谢谢。




bmdxyn0725 (<https://www.acwing.com/user/myspace/index/11886/>) 8个月前 回复 AcWing 790: 数的三次方根 这个模版, 为什么check(mid)后, l 与 r 的更新都是mid呢? (<https://www.acwing.com/user/myspace/index/11886/>)

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 8个月前 回复 (<https://www.acwing.com/user/myspace/index/11886/>)

因为mid不是整数, 实数可以取任意值, 不存在中点取不到的情况。



eric08 (<https://www.acwing.com/user/myspace/index/10819/>) 9个月前 回复 为什么快排在 相等的情况下也要交换 (<https://www.acwing.com/user/myspace/index/10819/>)

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 8个月前 回复 (<https://www.acwing.com/user/myspace/index/10819/>)

就算法而言, 这里交不交换都是正确的, 但这里在相等时也停止可以让代码简洁一些。




AYX (<https://www.acwing.com/user/myspace/index/11058/>) 9个月前 回复 关于快排的模板: (<https://www.acwing.com/user/myspace/index/11058/>)


`int i = l - 1, j = r + 1,`

左边指针 i 为啥要-1, 如果 $l == 0$, 那 i 岂不是变成-1了?


同样, r 为啥要+1?

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复 (<https://www.acwing.com/user/myspace/index/11058/>)

每次递归会先加1, 再判断, 同样也是先减1再判断。

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复 (<https://www.acwing.com/user/myspace/index/11058/>)


所以就将最开始的 l 和 r 都减成 $l-1$ 和 $r-1$ 。

 AYX (<https://www.acwing.com/user/myspace/index/11058/>) 9个月前 回复了 yxc 的评论 回复 (<https://www.acwing.com/user/myspace/index/11058/>)

明白了

另外发现了一个网站bug哈, 在手机上打开没法回复~~


期待啥时候搞个app, 以方便大家利用碎片时间学习

 AYX (<https://www.acwing.com/user/myspace/index/11058/>) 9个月前 回复了 yxc 的评论 回复 (<https://www.acwing.com/user/myspace/index/11058/>)


另外, 还发现一个特别奇妙的地方, 就是`quick_sort(q, l, j)`, `quick_sort(q, j + 1, r)`, 用 j , $j+1$ 来划分, 为啥这里不用 i 来划分?

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复了 AYX 的评论 回复 (<https://www.acwing.com/user/myspace/index/1/>)


手机是什么系统, 以及用的是哪个浏览器啊, 可能是有些语法不兼容。

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复了 AYX 的评论 回复 (<https://www.acwing.com/user/myspace/index/1/>)


也可以不用 $q[l]$ 来划分, 但分界点也需要相应变一下, 不能使用 $q[l]$, 因为当区间长度是2的时候, 如果 $x = q[l]$, 并且 $q[l] < q[r]$, 就会无限递归。

 AYX (<https://www.acwing.com/user/myspace/index/11058/>) 9个月前 回复了 yxc 的评论 回复 (<https://www.acwing.com/user/myspace/index/11058/>)


iOS, 自带的safari

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复了 AYX 的评论 回复 (<https://www.acwing.com/user/myspace/index/1/>)

好滴, 我去研究一下。

 寒星吻月 (<https://www.acwing.com/user/myspace/index/38498/>) 1个月前 回复了 yxc 的评论 回复 (<https://www.acwing.com/user/myspace/index/38498/>)

这里应该是 $q[l] > q[r]$ 才会无限递归。。

 yxc (<https://www.acwing.com/user/myspace/index/1/>) 18天前 回复了 寒星吻月 的评论 回复 (<https://www.acwing.com/user/myspace/index/1/>)

不是的, `quick_sort`时会直接`return`, 无限递归产生的原因是当区间长度为2时, 划分成的两个区间的长度分别是0和2, 就无限递归了。



Vodka编程菜菜 (<https://www.acwing.com/user/myspace/index/10353/>) 9个月前 回复 (<https://www.acwing.com/user/myspace/index/10353/>)

```

int partition(vector<int>& input, int l, int r) {
    int v = input[r];
    int s = l, e = r - 1;
    while(s <= e) {
        while(s <= e && input[s] <= v) s++;
        while(s <= e && input[e] >= v) e--;
        if(s <= e) swap(input[s], input[e]);
    }
    swap(input[s], input[r]);
    return s;
}

void quicksort(vector<int>& input, int l, int r) {
    if (l >= r) return;
    int pivot = partition(input, l, r);
    //cout << pivot << endl;
    quicksort(input, l, pivot - 1);
    quicksort(input, pivot + 1, r);
    return;
}

```

这个虽然罗嗦感觉更适合面试使用。

yx (https://www.acwing.com/user/myspace/index/1/) 9个月前 回复
(https://www.acwing.com/user/myspace/index/1/)

北极熊问企鹅为啥不去找它玩 (https://www.acwing.com/user/myspace/index/14215/) 3个
月前 回复 yx 的评论 回复 (https://www.acwing.com/user/myspace/index/14215/)

```

int partition(vector<int>& a, int l, int r)
{
    int v = a[l + r >> 1];
    swap(a[l + r >> 1], a[r]);
    int s = l, e = r - 1;
    while(s <= e)
    {
        while(s <= e && a[s] <= v) s++;
        while(s <= e && a[e] >= v) e--;
        if(s <= e) swap(a[s], a[e]);
    }
    swap(a[s], a[r]);
    return s;
}

void quicksort(vector<int>& a, int l, int r)
{
    if(l >= r) return;
    int p = partition(a, l, r);
    quicksort(a, l, p - 1);
    quicksort(a, p + 1, r);
}

```

按上面这位的写法，我把分界点变成了中点，可是还是要超时是怎么回事捏？

yx (https://www.acwing.com/user/myspace/index/1/) 3个月前 回复了 北极熊问企鹅为啥不去
(https://www.acwing.com/user/myspace/index/1/)

这种写法当所有数均相同时会变成 $O(n^2)$ 的复杂度。

北极熊问企鹅为啥不去找它玩 (https://www.acwing.com/user/myspace/index/14215/) 3个
月前 回复 yx 的评论 回复 (https://www.acwing.com/user/myspace/index/14215/)

啊对的！大佬太喜欢你这回复的效率了！

北极熊问企鹅为啥不去找它玩 (https://www.acwing.com/user/myspace/index/14215/) 3个
月前 回复 yx 的评论 回复 (https://www.acwing.com/user/myspace/index/14215/)

大佬所以你的写法在任何时候都不会变成 $O(n^2)$ 是嘛？

yx (https://www.acwing.com/user/myspace/index/1/) 3个月前 回复了 北极熊问企鹅为啥不去
(https://www.acwing.com/user/myspace/index/1/)

也是会的。不过被出题人卡的概率不大。

Vodka编程菜菜 (https://www.acwing.com/user/myspace/index/10353/) 9个月前 回复
这模板总结的好！不过我更倾向于根据这个帖子总结出自己的模板。
(https://www.acwing.com/user/myspace/index/10353/)

yx (https://www.acwing.com/user/myspace/index/1/) 9个月前 回复

(<https://www.acwing.com/user/myspace/index/1/>)



goontry (<https://www.acwing.com/user/myspace/index/4652/>) 9个月前 回复

aaa, y总快排中while循环内, 缺少else。
(<https://www.acwing.com/user/myspace/index/4652/>)



yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复

(<https://www.acwing.com/user/myspace/index/1/>) 快排模板有很多do-while给出的模板可以不用写else~



goontry (<https://www.acwing.com/user/myspace/index/4652/>) 9个月前 回复了 yxc 的评论

(<https://www.acwing.com/user/myspace/index/4652/>)

不写else, 会tle的。试了好久[jiong]



yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复了 goontry 的评论 回复

(<https://www.acwing.com/user/myspace/index/1/>) 你说的带else的版本在哪



goontry (<https://www.acwing.com/user/myspace/index/4652/>) 9个月前 回复了 yxc 的评论

(<https://www.acwing.com/user/myspace/index/4652/>)

hhaa, y总抱歉, 我刚反应过来。又试了一下, 发现是我代码写错了。



yxc (<https://www.acwing.com/user/myspace/index/1/>) 9个月前 回复了 goontry 的评论 回复

(<https://www.acwing.com/user/myspace/index/1/>) 好的