

太戈编程
etiger.vip

信奥算法

模拟问题
算法优化

现场挑战

321

难点分析

2个关键步骤	挑选未标记数里最小的数字进行标记
	所有未标记正数减1。
最值问询	找剩余数字里最小值
全面更新	原题要求修改每个数
	每个数都减1,不影响最小值是哪个
	所以不需要真的更新

代码1: multiset

```
4  int n,x,cnt=0;
5  cin>>n;
6  multiset<int> ms;
7  for(int i=1;i<=n;i++){
8      cin>>x;
9      ms.insert(x);
10 }
11 for(int t=0;t<n;t++){
12     x=*ms.begin();
13     if(x>t) cnt++;
14     
15 }
16 cout<<cnt<<endl;
```

代码1: multiset

```
4  int n,x,cnt=0;
5  cin>>n;
6  multiset<int> ms;
7  for(int i=1;i<=n;i++){
8      cin>>x;
9      ms.insert(x);
10 }
11 for(int t=0;t<n;t++){
12     x=*ms.begin();
13     if(x>t) cnt++;
14     ms.erase(ms.begin());
15 }
16 cout<<cnt<<endl;
```

跟着老师翻译
理解每一行

代码2: 数组

```
3  const int N=109;  
4  int x[N];  
5  int main(){  
6      int n,cnt=0;  
7      cin>>n;  
8      for(int i=0;i<n;i++)cin>>x[i];  
9      sort(x,x+n);  
10     for(int i=0;i<n;i++)  
11         if() cnt++;  
12     cout<<cnt<<endl;  
13     return 0;  
14 }
```

代码2: 数组

```
3  const int N=109;  
4  int x[N];  
5  int main(){  
6      int n,cnt=0;  
7      cin>>n;  
8      for(int i=0;i<n;i++)cin>>x[i];  
9      sort(x,x+n);  
10     for(int i=0;i<n;i++)  
11         if(x[i]>i) cnt++;  
12     cout<<cnt<<endl;  
13     return 0;  
14 }
```

数组x排序后

统计有几个数
大于自己编号

跟着老师翻译
理解每一行

讨论

为什么这题用数组
实现很方便

因为静态数据不会变化
找最值可以提前排序

multiset/set
处理动态问题才有优势

现场挑战

320

手算样例

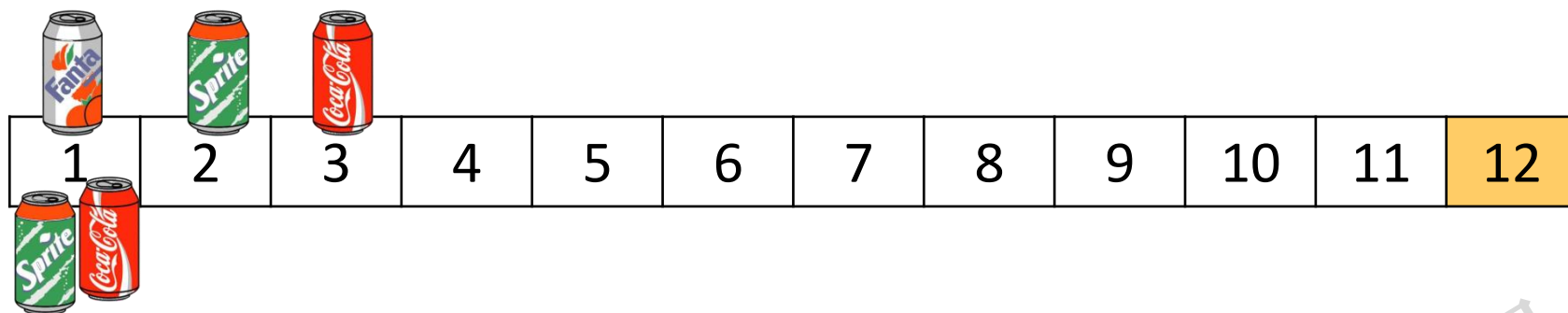
输入

5 12

1 1 2 3 1

输出

4



手算样例

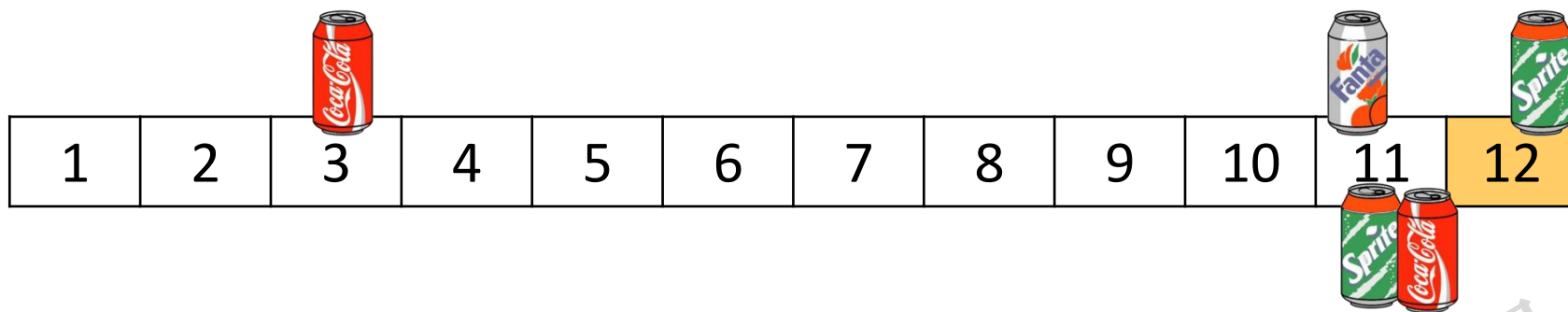
输入

5 12

1 1 2 3 1

输出

4



算法：模拟

模拟
每一步

题目说怎么操作就跟着操作

方法
讨论

用数组储存易拉罐坐标
 $x[1], x[2], \dots, x[n]$

难点

每次找到最小值后
需要更新该数值

动态
更新

`multiset/set`
处理动态问题有优势

代码1: multiset

跟着老师翻译
理解每一行

```
4  int n,m;  
5  cin>>n>>m;  
6  multiset<int> ms;  
7  for(int i=0;i<n;i++){  
8      int x;  
9      cin>>x;  
10     if(x>=m){cout<<0<<endl;return 0;}  
11     if(x>=0) ms.insert(x);  
12 }
```

代码1: multiset

```
13 int cnt=0;
14 while(1){
15     cnt++;
16     int p=*ms.begin();
17     if(p+10>=m) break;
18     ms.erase(ms.begin());
19     ms.insert(p+10);
20 }
21 cout<<cnt<<endl;
```

跟着老师翻译
理解每一行

能否
加速

时间复杂度
 $O(N\log N)$

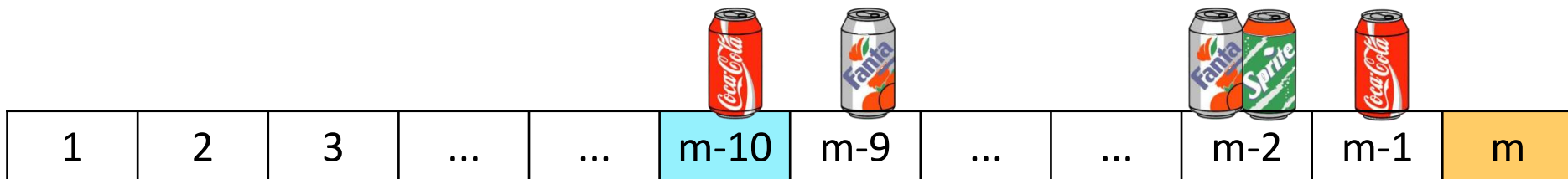
数学分析

终局
思维

在准备踢最后**1**脚的时候
大量易拉罐聚集在终点前的区域

坐标范围 $m-10\dots, m-2, m-1$

此时再踢**1**脚就一定到终点线



数学分析

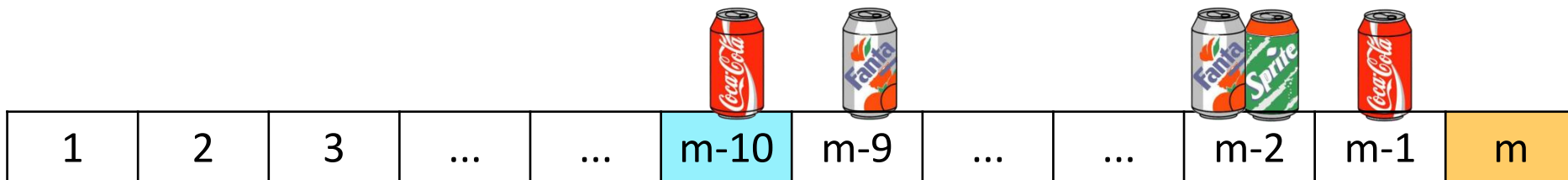
终局
思维

在准备踢最后1脚的时候
大量易拉罐聚集在终点前的区域

终点前
区域

坐标范围 $m-10, \dots, m-2, m-1$

此时再踢1脚就一定到终点线

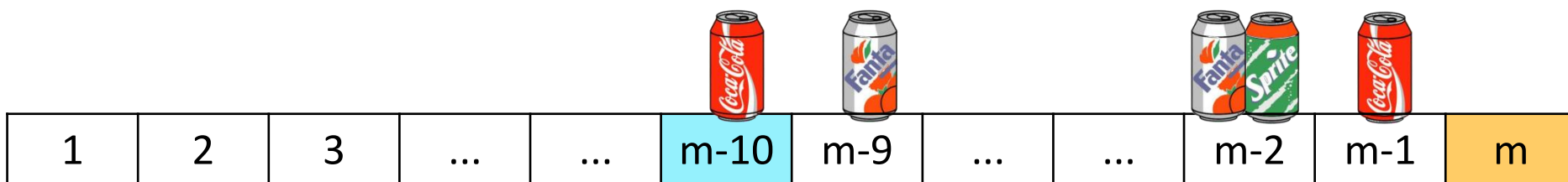


如何求
次数k

设起点为x的易拉罐要踢k脚
恰能进入 $\{m-10, m-9, \dots, m-1\}$?

数学分析

$$m-10 \leq x+10*k \leq m-1$$



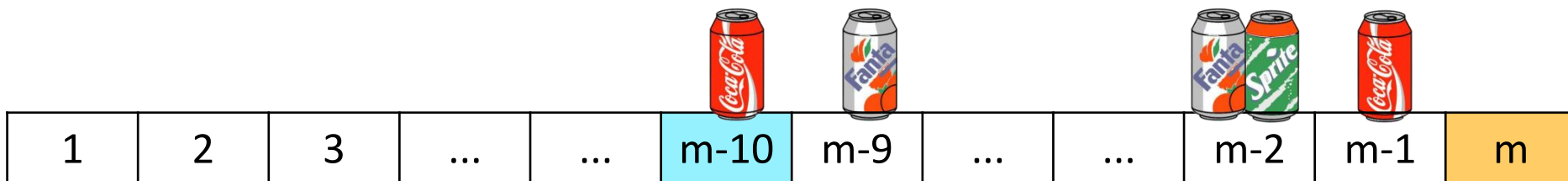
如何求
次数k

设起点为x的易拉罐要踢k脚
恰能进入 $\{m-10, m-9, \dots, m-1\}$?

数学分析

$$m-10 \leq x+10*k \leq m-1$$

$$m-10-x \leq 10*k \leq m-1-x$$



如何求
次数k

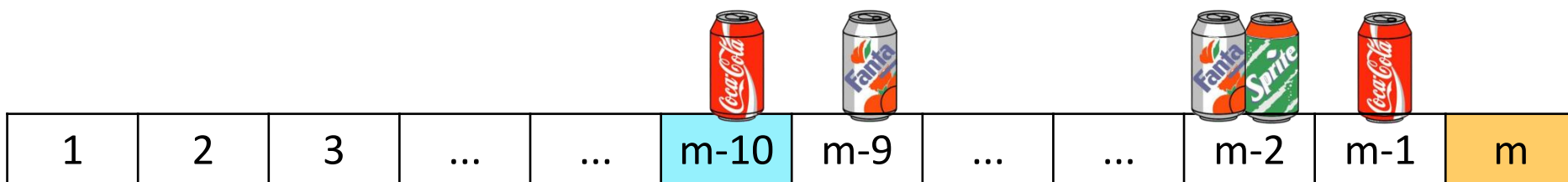
设起点为x的易拉罐要踢k脚
恰能进入 $\{m-10, m-9, \dots, m-1\}$?

数学分析

$$m-10 \leq x+10*k \leq m-1$$

$$m-10-x \leq 10*k \leq m-1-x$$

$$(m-x)/10.0-1 \leq k \leq (m-1-x)/10.0$$



如何求
次数k

设起点为x的易拉罐要踢k脚
恰能进入 $\{m-10, m-9, \dots, m-1\}$?

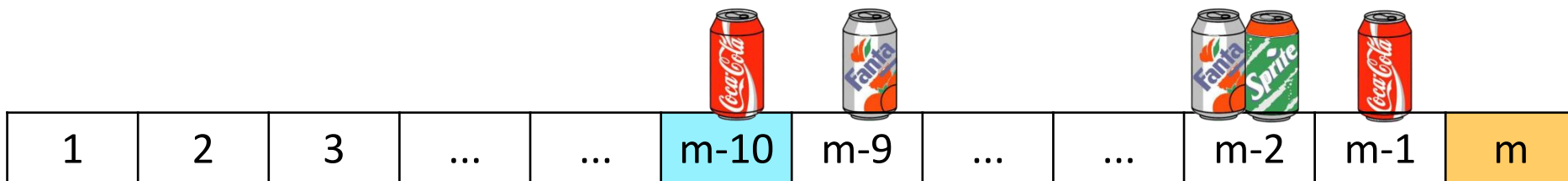
数学分析

$$m-10 \leq x+10*k \leq m-1$$

$$m-10-x \leq 10*k \leq m-1-x$$

$$(m-x)/10.0-1 \leq k \leq (m-1-x)/10.0$$

C++里
用/可取整



如何求
次数k

设起点为x的易拉罐要踢k脚
恰能进入 $\{m-10, m-9, \dots, m-1\}$?

代码2: 数学方法

```
4  int n,m,ans=1;
5  cin>>n>>m;
6  for(int i=0;i<n;i++){
7      int x;
8      cin>>x;
9      if(x>=m){cout<<0<<endl;return 0;}
10     if(x>=0)
11         ans+=(m-1-x)/10;
12 }
13 cout<<ans<<endl;
```

时间复杂度
 $O(N)$

现场挑战

342

代码1 数组

```
7  int m,n,cnt=0,ans=0;
8  cin>>m>>n;
9  for(int i=0;i<n;i++){
10     int x;
11     cin>>x;
12     if(ok[x])continue;
13     ans++;
14     ok[x]=1;
15     if(cnt<m){
16         id[cnt]=i;
17         value[cnt]=x;
18         cnt++;
19     }else{
20         int k=min_element(id,id+m)-id;
21         ok[value[k]]=0;
22         id[k]=i;
23         value[k]=x;
24     }
25 }
```

cnt: 内存里已有几个数

ok[x]: x是否在内存里

id[k]: 内存k号数原来是几号

value[k]: 内存k号的数值

跟着老师翻译
理解每一行

代码2 数组

```
7  int m,n,cnt=0,ans=0;
8  cin>>m>>n;
9  for(int i=0;i<N;i++)pos[i]=INF;
10 for(int i=0;i<n;i++){
11     int x;
12     cin>>x;
13     if(pos[x]<INF)continue;
14     ans++;
15     pos[x]=i;
16     if(cnt<m)cnt++;
17     else{
18         int k=min_element(pos,pos+N)-pos;
19         pos[k]=INF;
20     }
21 }
```

pos[x]==INF 代表x不在内存里

pos[x]代表x进入内存时原编号

cnt: 内存里已有几个数

跟着老师翻译
理解每一行

```
7  int m,n,cnt=0,ans=0;
8  cin>>m>>n;
9  for(int i=0;i<N;i++)pos[i]=INF;
10 for(int i=0;i<n;i++){
11     int x;
12     cin>>x;
13     if(pos[x]<INF)continue;
14     ans++;
15     pos[x]=i;
16     if(cnt<m)cnt++;
17     else
18         *min_element(pos,pos+N)=INF;
19 }
```

3 **const int** N=1009;

4 **int** pos[N];

pos[x]代表x进入内存时原编号

5 **struct** info{

6 **int** v;

v代表数字大小

7 **bool** operator<(const info&b)const{

8 **return** pos[v]<pos[b.v];

9 }

10 };

pos[v]代表内存里数字v
进入内存时的先后位置

info的比较规则依赖于
外部信息pos[]数组

```
14 set<info> s;  
15 for(int i=1;i<=n;i++){  
16     info x;  
17     cin>>x.v;  
18     if(s.count(x)==0){  
19         ans++;  
20         if(s.size()==m) s.erase(s.begin());  
21         pos[x.v]=i;  
22         s.insert(x);  
23     }  
24 }
```

跟着老师翻译
理解每一行

太戈编程

321. 伤员输血

320. 足球爱好者

342. 机器翻译

拓展题

310. 炮轰僵尸

317. 数据流中位数