

NOIP2012借教室

2014年4月4日

1,507

1

题目描述

在大学期间，经常需要租借教室。大到院系举办活动，小到学习小组自习讨论，都需要向学校申请借教室。教室的大小功能不同，借教室人的身份不同，借教室的手续也不一样。

面对海量租借教室的信息，我们自然希望编程解决这个问题。

我们需要处理接下来 n 天的借教室信息，其中第 i 天学校有 r_i 个教室可供租借。共有 m 份订单，每份订单用三个正整数描述，分别为 d_j, s_j, t_j ，表示某租借者需要从第 s_j 天到第 t_j 天租借教室（包括第 s_j 天和第 t_j 天），每天需要租借 d_j 个教室。

我们假定，租借者对教室的大小、地点没有要求。即对于每份订单，我们只需要每天提供 d_j 个教室，而它们具体是哪些教室，每天是否是相同的教室则不用考虑。

借教室的原则是先到先得，也就是说我们要按照订单的先后顺序依次为每份订单分配教室。如果在分配的过程中遇到一份订单无法完全满足，则需要停止教室的分配，通知当前申请人修改订单。这里的无法满足指从第 s_j 天到第 t_j 天中有至少一天剩余的教室数量不足 d_j 个。

现在我们需要知道，是否会有订单无法完全满足。如果有，需要通知哪一个申请人修改订单。

输入

输入文件为 `classroom.in`。

第一行包含两个正整数 n, m ，表示天数和订单的数量。

第二行包含 n 个正整数，其中第 i 个数为 r_i ，表示第 i 天可用于租借的教室数量。

接下来有 m 行，每行包含三个正整数 d_j, s_j, t_j ，表示租借的数量，租借开始、结束分别在第几天。

每行相邻的两个数之间均用一个空格隔开。天数与订单均用从1开始的整数编号。

输出

输出文件为 `classroom.out`。

如果所有订单均可满足，则输出只有一行，包含一个整数 0。否则（订单无法完全满足）输出两行，第一行输出一个负整数-1，第二行输出需要修改订单的申请人编号。

样例输入

```
4 3 2 5 4 3 2 1 3 3 2 4 4 2 4
```

样例输出

```
-1 2
```

提示

【输入输出样例说明】

第 1 份订单满足后，4 天剩余的教室数分别为 0，3，2，3。第 2 份订单要求第 2 天到第 4 天每天提供 3 个教室，而第 3 天剩余的教室数为 2，因此无法满足。分配停止，通知第 2 个申请人修改订单。

【数据范围】

对于 10% 的数据，有 $1 \leq n, m \leq 10$ ；

对于 30% 的数据，有 $1 \leq n, m \leq 1000$ ；

对于 70% 的数据，有 $1 \leq n, m \leq 10^5$ ；

对于 100% 的数据，有 $1 \leq n, m \leq 10^6, 0 \leq r_i, d_j \leq 10^9, 1 \leq s_j \leq t_j \leq n$ 。

题解

线段树70分，裸的线段树操作

正解二分。。。似乎是这样的

用某一天的前缀和表示该天需要的教室数

比如一开始数列a是0 0 0 0 0 0

前缀和0 0 0 0 0 0

3到5天需要2的教室

将a[3]+=2,a[6]-=2

数列变为0 0 2 0 0 -2

前缀和变为0 0 2 2 2 0

这样就实现了增加3-5需要的教室数

然后我们二分订单数

处理前mid个订单看看是否有某一天的前缀和大于di

70分

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int n,m,ans,a[1000010],d[1000010],x[1000010],y[1000010],s[1000010],sum;
bool judge(int v)
{
    memset(s,0,sizeof(s));sum=0;
    for(int i=1;i<=v;i++)
    {s[x[i]]+=d[i];s[y[i]+1]-=d[i];}
```

```

11     for(int i=1;i<=n;i++)
12     {
13         sum+=s[i];
14         if(sum>a[i])return 0;
15     }
16     return 1;
17 }
18 int main()
19 {
20     scanf("%d%d",&n,&m);
21     for(int i=1;i<=n;i++)scanf("%d",&a[i]);
22     for(int i=1;i<=m;i++)
23         scanf("%d%d%d",&d[i],&x[i],&y[i]);
24     int l=1,r=m;
25     while(l<=r)
26     {
27         int mid=(l+r)>>1;
28         if(!judge(mid)){ans=mid;r=mid-1;}
29         else l=mid+1;
30     }
31     if(!ans)printf("0");
32     else printf("-1\n%d",ans);
33     return 0;
34 }

```

100分

```

1  #include<iostream>
2  #include<cstdio>
3  #include<cstring>
4  using namespace std;
5  int n,m,ans,a[1000010],d[1000010],x[1000010],y[1000010],s[1000010],sum;
6  template <class T> T get(T &u){
7      char x;for(;!isdigit(x=getchar()));
8      for(u=x-48;isdigit(x=getchar());u*=10,u+=(x-48));
9      ungetc(x,stdin);return u;
10 }
11 bool judge(int v)
12 {
13     memset(s,0,sizeof(s));sum=0;
14     for(int i=1;i<=v;i++)
15         {s[x[i]]+=d[i];s[y[i]+1]-=d[i];}
16     for(int i=1;i<=n;i++)
17     {
18         sum+=s[i];
19         if(sum>a[i])return 0;
20     }
21     return 1;
22 }
23 int main()
24 {
25     get(n),get(m);
26     for(int i=1;i<=n;i++)get(a[i]);
27     for(int i=1;i<=m;i++)
28         get(d[i]),get(x[i]),get(y[i]);
29     int l=1,r=m;
30     while(l<=r)
31     {
32         int mid=(l+r)>>1;
33

```

```
34     if(!judge(mid)) {ans=mid;r=mid-1;}
35     else l=mid+1;
36 }
37 if(!ans)printf("0");
38 else printf("-1\n%d",ans);
39 return 0;
}
```