

题意

公元2411年，人类开始在地球以外的行星建立居住点。在第1326号殖民星上， N 个居住点分布在一条直线上。为了方便描述，我们设第 i 个居住点的位置是 X_i ，其中居住着 Y_i 位居民。随着冬季的到来，一些人口较多的居住点的生态循环系统已经开始超负荷运转。为了顺利度过严冬，殖民星上的居民一致同意通过转移到人口较少的居住点来减轻人口众多的居住点的负荷。

遗憾的是，1326殖民星的环境非常恶劣。在冬季到来前，每个居民点的居民最远能迁移到距离不超过 R 的居民点。1326殖民星的居民希望知道，如何安排迁移才能使完成迁移后人口最多的居民点人口最少？

注意有可能存在多个居民点位置相同。

分析

最多的居民点人口最少 —— 二分答案

check方法：

贪心将每个人往当前能迁移的最靠前的地方迁移

每个居住点，维护两个值 —— 原著、迁移来的人

双指针维护，当前要迁移的居住点 now ，和最靠前的没满的居住点 pre （该点以前都满了）

将 now 的原著往 pre 迁移，如果 now 原著空了，则 $now++$ ，如果 pre 迁移的人满了，则

$pre++$ ，对于留在原地的居民，可以视作迁移到自己的城市

如果 pre 在 now 的后面，并且距离大于 R 了，那么 now 点的人迁移不完了，返回 $false$

如果最后还有原著没有迁移，返回 $false$

否则返回 $true$

代码

```
1 #include<algorithm>
2 #include<bitset>
3 #include<cstdio>
4 #include<cstring>
5 #include<cstdlib>
6 #include<cmath>
7 #include<deque>
8 #include<iostream>
9 #include<map>
10 #include<queue>
11 #include<set>
12 #include<stack>
13 #include<string>
14 #include<vector>
```

```

15 #include<list>
16 #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
17 #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
18 #define Memset(a,b); memset((a),(b),sizeof((a)));
19 #define Cout(a,b); printf("%d", (a));printf(b);
20 #define Coutc(a,b); printf("%c", (a));printf(b);
21 #define Couts(a,b); printf("%s", (a));printf(b);
22 using namespace std;
23 const int INF = 0x3f3f3f3f;
24 typedef long long LL;typedef unsigned long long ULL;typedef long double
    LDB;
25 inline LL CinLL(){LL x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){if(ch=='-'
    ')f=-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x*10+ch-
    '0';ch=getchar();}return x*f;}
26 inline int Cin(){int x=0,f=1;char ch=getchar();while(!isdigit(ch)){if(ch=='-'
    ')f=-1;ch=getchar();}while(isdigit(ch))x=x*10+ch-'0',ch=getchar();return f*x;}
27 int n,r;
28 struct cre{
29     LL x,num;
30 }a[100005];
31 bool cmp(cre p,cre q){
32     return p.x<q.x;
33 }
34 int book[100005][2];
35 int check(int aim)
36 {
37     Memset(book,0);
38     int pre = 1,now = 1;
39     For(i,1,n) book[i][0] = a[i].num;
40     while(now<=n)
41     {
42
43         if(abs(a[now].x - a[pre].x) > r){
44             if(pre > now) return 0;
45             else pre++;
46         }
47         else {
48             if(book[pre][1] + book[now][0] <= aim){
49                 book[pre][1] += book[now][0];
50                 book[now][0] = 0;
51                 now++;
52             }
53             else{
54
55                 book[now][0] -= (aim-book[pre][1]);
56                 book[pre][1] = aim;

```

```

57         pre++;
58     }
59 }
60 }
61 For(i,1,n) if(book[i][0]) return 0;
62 return 1;
63 }
64 int main()
65 {
66     ios::sync_with_stdio(false);
67     int _;
68     cin>>_;
69     while(_--)
70     {
71         cin>>n>>r;
72         LL l = 0 , r = 0;
73         For(i,1,n){
74             cin>>a[i].x>>a[i].num;
75             r = max(r,a[i].num);
76         }
77         sort(a+1,a+n+1,cmp);
78         while(l < r)
79         {
80             int mid = (l+r)>>1;
81             if(check(mid) == 1)
82                 r = mid;
83             else
84                 l = mid + 1;
85         }
86         cout<<r<<endl;
87     }
88 }

```