

题意

切水果游戏

- 给出n个水果在屏幕中出现的时间闭区间
- 每次切会完全清屏
- 同时切三个以上才计分，否则不计分
- 不能遗漏水果
- 水果数 $n \leq 1000$ 时间区间 $0 \leq x_i \leq y_i \leq 1000000000$

问最高得分

SAMPLE

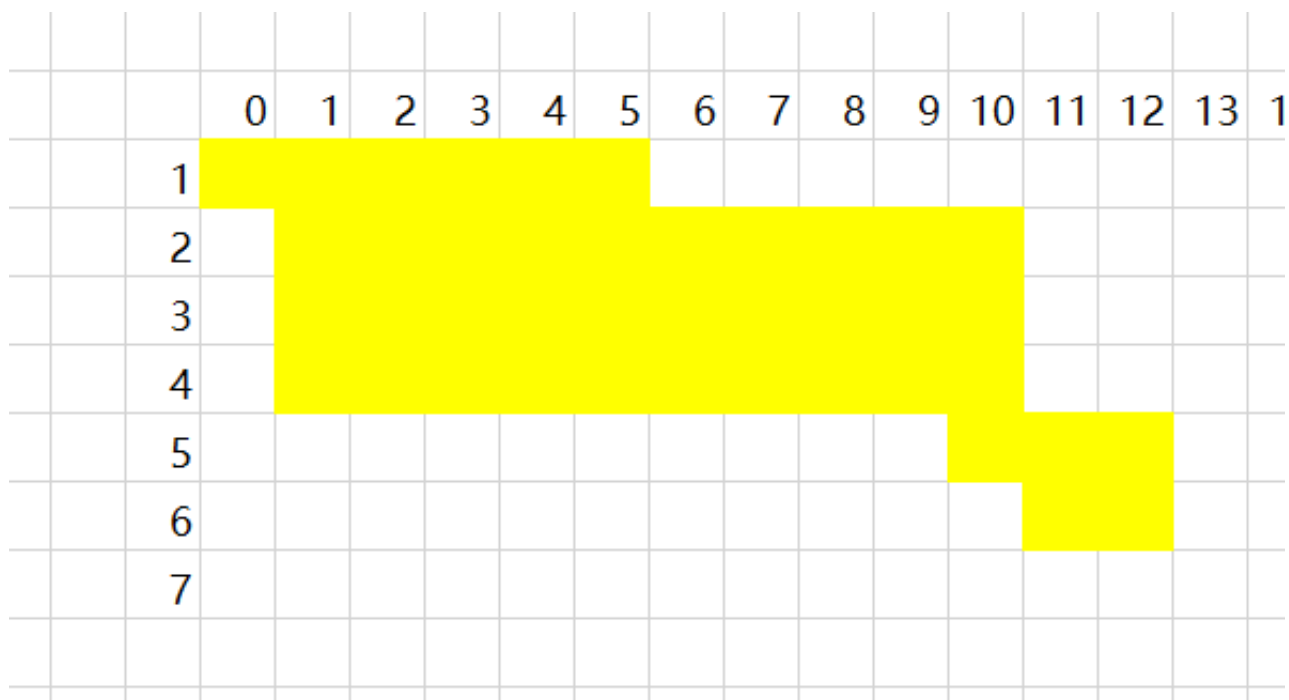
input	output
1	
10	
1 10	
2 11	
3 12	
4 13	
13 14	
14 15	
13 19	
20 22	
21 23	
22 24	Case #1: 10

思路分析

从每次切必须清屏入手

我需要确定 **动刀的时间**

由于区间范围太大（1000000000），而水果数量 最多只有1000
分析可知：



- 如果一个时间段 (L, R) 都没有新水果出现, 任意时间切效果都一样, 所以我们默认在可以切的点在新水果出现之前
- 将所有水果以左区间排序
- 遍历水果, 如果 $x[i] \neq x[i-1]$, 则说明时间点 $x[i]$ 有新水果出现, 则 $x[i-1]$ 是一个可以动刀的点 (转移点)

于是 设计dp方程

$dp[i]$: 第 i 个水果出现时 ($t = i[i]$) 切最后一刀

$dp[i] = \max(dp[i], dp[j-1] + \text{sum}(j, i))$ $\text{sum}(j, i)$ 是从第 j 个到第 i 个水果, 在时间 t 时能切到的个数 (如果小于3个 $\text{sum} = 0$)

最后答案为 $dp[n]$

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long LL;
4  struct cre{
5      int l,r;
6  };
7  bool cmp(cre x,cre y)
8  {
9      if(x.l == y.l) return x.r < y.r;
10     return x.l < y.l;
11 }
12 cre a[1005];
13 int dp[1005];
14 int main()
15 {

```

```

16 ios::sync_with_stdio(false);
17 int T,_ = 0;
18 cin >> T;
19 while(T--)
20 {
21     memset(dp,0,sizeof(dp));
22     int n;
23     cin >> n;
24     for(int i = 1;i <= n;i++)
25         cin >> a[i].l >> a[i].r;
26     sort(a+1,a+n+1,cmp);
27     for(int i = 1;i <= n;i++)
28     {
29         if(i < n && a[i].l == a[i+1].l) continue;
30         int sum = 0,count = 0;
31         for(int j = i;j >= 1;j--)
32         {
33             if(a[i].l <= a[j].r)
34                 count++;
35             if(count > 2)
36                 sum = count;
37             else
38                 sum = 0;
39             if(j == 1 || a[j].l != a[j-1].l)
40             {
41                 dp[i] = max(dp[i],dp[j - 1] + sum);
42             }
43         }
44     }
45     cout << "Case #" << ++_ << ": ";
46     cout<<dp[n]<<endl;
47 }
48 }

```