T3

1.【题目】有 n 只猎豹，第 i 只猎豹t_i时刻出发，以速度v_i每时刻向前奔跑。找一个时刻，使得所有猎豹都已经出发并且跑得最快的和最慢的猎豹之间距离尽可能小。求这个最小值。

20%　n=2

另 20%　n=3

60%　n<=100

80%　n<=1000

100%　n<=100000，1<=t_i，v_i<=100000

2.分析

求“最大的最小”，无疑是二分或三分了。很显然，随着时间的变化，跑得最快的和最慢的猎豹之间的距离是一个大致先单调递减后单调递增的函数，可以用三分来求最值。思路确定，后面就好做了。

首先，确定区间的左端点和右端点。因为题目中“所有猎豹都已经出发”，左端点只能是猎豹的最晚出发时间。对于右端点，可以根据数据范围求三分最多次数，进而确定。右端点最小为 1e6 就可以了。

其次，求对应时间的对应值。可以将这种对应关系设为函数 f(x)，要求 f(x)，只需枚举每一只猎豹，算出当时刻为 x 时所走的路程，再取个最小值和最大值，再用最大值减去最小值就可以得到在时刻为 x 时跑得最快的与最慢的猎豹的距离。(考试错的地方!!!)

最后，判断取舍情况。因为这是一个先单调递减后单调递增的函数，如果 f(lmid)<f(rmid)，那么舍去 rmid；否则，舍去左端点。

【注意】精度问题。一般在 1e-8 或 1e-10 即可。

结束，接下来开始码代码就可以了。

3. 代码

```
# include <cstdio>
# include <algorithm>
# define N 100010
# define eps 1e-11
using namespace std;
struct Node
{
    int t;
    int v;
} kkk[N];
int n;
double L,R,lmid,rmid;
double f(double x);
int main()
{
    //freopen("chase.in","r",stdin);
    //freopen("chase.out","w",stdout);
    scanf("%d",&n);
```

```
    for(int i=1;i<=n;i++)
    {
        scanf("%d%d",&kkk[i].t,&kkk[i].v);
        L=max(L,(double)kkk[i].t);
    }
    R=1e15;
    while(R-L>eps)
    {
        lmid=L+(R-L)/3;
        rmid=R-(R-L)/3;
        if(f(lmid)<f(rmid)) R=rmid;
        else L=lmid;
    }
    printf("%.2lf",f(L));
    return 0;
}
double f(double x)
{
    double minn,maxx,temp;
    minn=maxx=(x-kkk[1].t)*kkk[1].v;
    for(int i=2;i<=n;i++)
    {
        temp=(x-kkk[i].t)*kkk[i].v;
        minn=min(minn,temp);
        maxx=max(maxx,temp);
    }
    return maxx-minn;
}
```

AC 鼓掌!!!