

选择不相交区间

```
#include <bits/stdc++.h>
using namespace std;

struct qu{
    int l,r;
};
int cmp(qu x,qu y){
    return x.r<y.r;
}
qu a[1000005];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        scanf("%d %d",&a[i].l,&a[i].r);
    }
    sort(a+1,a+n+1,cmp);
    int t,ans=1;
    t=a[1].r;
    for(int i=2;i<=n;i++){
        if(a[i].l>=t){ //如果不冲突
            t=a[i].r; //选择这个区间，然后更新为这个区间的右端点
            ans++; //选择的区间数加1
        }
    }
    cout<<ans;
    return 0;
}
```

区间选点问题

```
#include <bits/stdc++.h>
using namespace std;

struct qu
{
    int a,b;
};

qu s[1000010];

int cmp(qu x,qu y)
{
    if(x.b<y.b) return 1;
    else if (x.b==y.b&& x.a<y.b) return 1;
    else return 0;
}

int main()
```

```

{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        scanf("%d %d",&s[i].a,&s[i].b);
    sort(s+1,s+1+n,cmp);

    int end,cnt=1;
    end=s[1].b;
    for(int i=2;i<=n;i++)
    {
        if(s[i].a>end) //如果当前的点无法选择区间s[i]
        {
            end=s[i].b; //更新为s[i]区间的右端点
            cnt++; //加一个点
        }
    }
    printf("%d\n",cnt);
    return 0;
}

```

区间覆盖问题

```

#include <bits/stdc++.h>
using namespace std;

struct qu
{
    int a,b;
};

qu ss[1000010];

int cmp(qu x,qu y)
{
    if(x.a<y.a) return 1;
    else return 0;
}

int main()
{
    int n,s,t;
    cin>>n;
    for(int i=1;i<=n;i++)
        scanf("%d %d",&ss[i].a,&ss[i].b);
    cin>>s>>t;
    sort(ss+1,ss+1+n,cmp);
    int cnt=0,flag=0,end=0;
    for(int i=1;i<=n;i++)
    {
        while(ss[i].a<=s&&i<=n) //循环找到右端点最大的区间覆盖目标区间
        {
            end=max(end,ss[i].b);
            i++;
        }
        cnt++; //区间加一
    }
}

```

```
s=end;
if(ss[i].a>end) //无解情况1
{
    flag=1;break;
}
i--;
if(i==n&&end<t) //无解情况2
{
    flag=1;break;
}
if(ss[i].b>=t) break; //覆盖完成，退出
}
if(flag==0)
    printf("%d\n",cnt);
else
    printf("no solution!");
return 0;
}
```