A：

一眼水题，果然T了，开始想法是用排序算法，cmp函数如下：

bool cmp(node a,node b){

if(a.s==b.s) return a.id<b.id;

return a.s>b.s;

}

优化算法：可以用两个数组存下每一场比赛的输赢情况，易见这两个数组与每一轮比赛对应，是有序的

代码如下：

#include <cstdio>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

struct node{

int s,id,p;

};

node a[200005],win[100005],lose[100005];

bool cmp(node a,node b){

if(a.s==b.s) return a.id<b.id;

return a.s>b.s;

}

int main(){

int n,r,q;

scanf("%d%d%d",&n,&r,&q);

for(int i=1;i<=2\*n;i++){

a[i].id=i;

scanf("%d",&a[i].s);

}

for(int i=1;i<=2\*n;i++) scanf("%d",&a[i].p);

sort(a+1,a+2\*n+1,cmp);

while(r--){

int j=1,k=1,l=1;

for(int i=1;i<=n;i++){

if(a[2\*i-1].p>a[2\*i].p){

a[2\*i-1].s++;

win[j++]=a[2\*i-1];

lose[k++]=a[2\*i];

}

else{

a[2\*i].s++;

win[j++]=a[2\*i];

lose[k++]=a[2\*i-1];

}

}//记录输赢情况

int pos1=0,pos2=0;

for(int i=1,j=1;i<=n&&j<=n;){

if(cmp(win[i],lose[j])) a[l++]=win[i],i++;

else a[l++]=lose[j],j++;

//取出比赛信息，判断

pos1=i;

pos2=j;

}

while(pos2<=n) a[l++]=lose[pos2++];

while(pos1<=n) a[l++]=win[pos1++];

//将不需要比较的情况放入数组

//sort(a+1,a+2\*n+1,cmp);

}

printf("%d\n",a[q].id);

}

B：

太简单，没有题解

C：

找单词，暴力就好，但是空格的位置很坑

代码如下：

#include <cstdio>

#include <string>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

string a,b;

int judge(int pos){

for(int i=0;i<a.length();i++)

if(b[pos+i]!=a[i]) return 0;

if(b[pos+a.length()]!=' ') return 0;

//这里是避免单词被包含的情况

return 1;

}

int main(){

getline(cin,a);

getline(cin,b);

for(int i=0;i<a.length();i++)

if(a[i]>='A'&&a[i]<='Z') a[i]=a[i]-'A'+'a';

for(int i=0;i<b.length();i++)

if(b[i]>='A'&&b[i]<='Z') b[i]=b[i]-'A'+'a';

//转换为小写

int cnt=0,pos=-1;

b=" "+b;

//加一个空格是为了更好的与判断的格式配合，如不加空格

//则需要手动判断第一个字符的情况

for(int i=0;i<b.length();i++)

if(b[i]==' '&&b[i+1]!=' '){

//以空格为开始标志，这里也有其他的判断方式，但这种方法应更简单

if(judge(i+1)){

cnt++;

if(pos==-1) pos=i;

}

}

if(cnt==0) printf("-1");

else printf("%d %d",cnt,pos);

}

E：

这里需要用二项式定理，我用到了杨辉三角求解二项式系数，学校培训课讲过了

代码如下：

#include <cstdio>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

int c[1005];

int qpow(int x,int y){

int ans=1;

while(y>0){

if(y&1) ans\*=x,ans%=10007;

x\*=x;

x%=10007;

y/=2;

}

return ans;

}

int main(){

int a,b,k,n,m;

scanf("%d%d%d%d%d",&a,&b,&k,&n,&m);

int ans=qpow(a%10007,n)\*qpow(b%10007,m)%10007;

c[1]=1,c[2]=1;

for(int i=2;i<k;i++)

for(int j=i+1;j>=1;j--) c[j]=(c[j]+c[j-1])%10007;

ans=(c[min(n,m)]+c[min(n,m)+1])%10007\*ans;

printf("%d\n",ans%10007);

}

H：

暴力判断就好，没什么讲的

代码如下：

#include <cstdio>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

int x[10005],y[10005],w[10005],h[10005];

int main(){

int n;

scanf("%d",&n);

for(int i=1;i<=n;i++) scanf("%d%d%d%d",&x[i],&y[i],&w[i],&h[i]);

int a,b;

scanf("%d%d",&a,&b);

for(int i=n;i>=1;i--)

if(x[i]<=a&&y[i]<=b)

if(x[i]+w[i]>=a&&y[i]+h[i]>=b){

printf("%d\n",i);

return 0;

}

printf("-1\n");

}

I：

此题解法较多，先枚举符合最低消费的咖啡馆，统计答案，若当前咖啡馆i合法，则答案ans+=cnt2[x]\*cnt1[x]+a[i][0]==x?cnt2[x]-1:0

但此处会有重复计算的情况，cnt2[x]表示i之前的合法点数，cnt[1]表示i之后的点数，而我们只需要计算一次即可，所以经过一个合法节点，cnt2清0

代码如下：

#include <cstdio>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

int a[300001][2],cnt1[51],cnt2[51];

int main(){

int n,k,p;

scanf("%d%d%d",&n,&k,&p);

for(int i=1;i<=n;i++){

scanf("%d%d",&a[i][0],&a[i][1]);

cnt1[a[i][0]]++;

}

long long ans=0;

for(int i=1;i<=n;i++){

cnt2[a[i][0]]++;

cnt1[a[i][0]]--;

if(a[i][1]<=p){

for(int x=0;x<=k;x++){

if(x==a[i][0]) ans+=cnt2[x]\*cnt1[x]+cnt2[x]-1;

else ans+=cnt1[x]\*cnt2[x];

cnt2[x]=0;

}

//核心代码，代码写的很清楚了，注释没有可写的了，注意理解

}

}

printf("%lld\n",ans);

}