**求最大值**

问题描述

　　给n个有序整数对ai bi，你需要选择一些整数对 使得所有你选定的数的ai+bi的和最大。并且要求你选定的数对的ai之和非负，bi之和非负。

输入格式

　　输入的第一行为n，数对的个数  
　　以下n行每行两个整数 ai bi

输出格式

　　输出你选定的数对的ai+bi之和

样例输入

5  
-403 -625  
-847 901  
-624 -708  
-293 413  
886 709

样例输出

1715

数据规模和约定

　　1<=n<=100  
　　-1000<=ai,bi<=1000

C++代码：

#include<cstdio>

#include<algorithm>

#include<cstring>

class Node {

public:

int as, bs, sum;

Node() { as = bs = sum = 0; }

};

Node arr[110];

Node dp[110];

Node total;

Node ret;

bool vis[110];

bool cmp(Node a, Node b) {

return a.sum>b.sum;

}

int Solve(int n) {

memset(vis, true, sizeof(vis));

std::sort(arr,arr+n,cmp);

/\*

for (int i = 0; i < n; ++i)

printf("%d %d %d\n", arr[i].as, arr[i].bs, arr[i].sum);

\*/

ret = arr[0];

bool update = true;

while (update) {

update = false;

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

if (ret.as + arr[i].as >= 0 && ret.bs + arr[i].bs >= 0 && arr[i].sum >= 0 && vis[i])

{

//printf("%d %d %d\n", arr[i].as, arr[i].bs, arr[i].sum);

vis[i] = false, update = true, ret.as += arr[i].as, ret.bs += arr[i].bs, ret.sum += arr[i].sum;

}

}

return ret.as>=0&&ret.bs>=0?ret.sum:0;

}

int main() {

//freopen("D://TestData/蓝桥/求最大值/input7.txt","r",stdin);

int n;

scanf("%d", &n);

int len = 0;

for (int i = 0; i < n; ++i) {

scanf("%d%d", &arr[len].as, &arr[len].bs),arr[len].sum=arr[len].as+arr[len].bs;

if (arr[len].as>0||arr[len].bs>0) {

total.as+=arr[len].as,total.bs+=arr[len].bs,total.sum+=arr[len].sum;

++len;

}

}

int ans = Solve(len);

if (total.as >= 0 && total.bs >= 0)printf("%d\n", std::max(ans,total.sum));

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

return 0;

}

C代码：

#include<stdio.h>

#include<stdlib.h>

int dp[100005];

int can[100005];

struct pair

{

int a, b;

} p[110];

int cmp(const void\* a, const void\* b)

{

return ((struct pair\*)b)->a - ((struct pair\*)a)->a;

}

int main(void)

{

int n, m;

int a, b;

int maxsum;

int suma, sumb, max;

int i, j;

scanf("%d", &n);

suma = 0;

sumb = 0;

for (i=0, m=0; i<n; ++i)

{

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

if (a>0 && b>0)

{

suma += a;

sumb += b;

}

else if (a<=0 && b<=0)

{

continue;

}

else

{

p[m].a = a;

p[m].b = b;

++m;

}

}

dp[suma] = sumb;

can[suma] = 1;

can[0] = 1;

qsort(p, m, sizeof(struct pair), cmp);

max = suma;

for (i=0; i<m&&p[i].a>0; ++i)

{

max += p[i].a;

for (j=max; j>=p[i].a; --j)

{

if (can[j-p[i].a]==1 && (can[j]==0 || dp[j]<dp[j-p[i].a]+p[i].b))

{

dp[j] = dp[j-p[i].a]+p[i].b;

can[j] = 1;

}

}

}

for (; i<m; ++i)

{

for (j=0; j<=max+p[i].a; ++j)

{

if (can[j-p[i].a]==1 && (can[j]==0 || dp[j]<dp[j-p[i].a]+p[i].b))

{

dp[j] = dp[j-p[i].a]+p[i].b;

can[j] = 1;

}

}

}

maxsum = 0;

for (i=0; i<=max; ++i)

{

if (can[i]==1&&dp[i]>=0&&maxsum<i+dp[i])

{

maxsum = i+dp[i];

}

}

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

return 0;

}

JAVA代码：

import java.util.\*;

public class Main{

public static void main(String[] args) {

int max=0,aa=0, bb=0;

boolean k=true;

Scanner in=new Scanner(System.in);

int n=in.nextInt();

int[] ai=new int[n];

int[] bi=new int[n];

for(int i=0;i<n;i++){

ai[i]=in.nextInt();

bi[i]=in.nextInt();

}

for(int i=0;i<n-1;i++){

for(int j=0;j<n-1;j++){

if((ai[j]+bi[j])<(ai[j+1]+bi[j+1])){

aa=ai[j];

ai[j]=ai[j+1];

ai[j+1]=aa;

bb=bi[j];

bi[j]=bi[j+1];

bi[j+1]=bb;

}

}

}

for(int i=0;i<n-1;i++){

for(int j=0;j<n-1;j++){

if(k==true){

if((ai[i]+bi[i])>=0||(ai[j+1]+bi[j+1])>=0){

if((ai[i]+ai[j+1]>=0) && (bi[i]+bi[j+1]>=0)){

aa=ai[i]+ai[j+1];

bb=bi[i]+bi[j+1];

max=aa+bb;

ai[i]=0;

ai[j+1]=0;

bi[i]=0;

bi[j+1]=0;

k=false;

}

}

}

else if((ai[j+1]+bi[j+1])>=0 ){

if( (ai[j+1]+aa>=0) && bi[j+1]+bb>=0){

aa+=ai[j+1];

bb+=bi[j+1];

max+=ai[j+1]+bi[j+1];

ai[j+1]=0;

bi[j+1]=0;

}

}

}

}

System.out.print(max);

}

}