算法实验10

问题:会场安排问题

代码: C++

#include<cstdio>

#include<cstdlib>

#include<algorithm>

using namespace std;

bool cmp(int a,int b)

{

return a<b;

}

int main()

{

int n,i;

int a[1000],b[1000];

scanf("%d",&n);

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

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

sort(a,a+n,cmp);

sort(b,b+n,cmp);

int j=0,sum=0;

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

{

if(a[i]<b[j])

sum++;

else

j++;

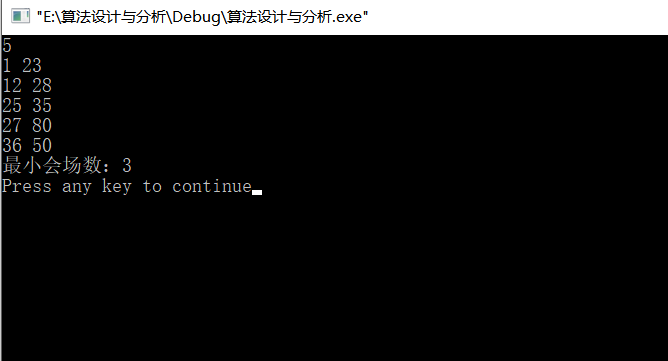
}

printf("最小会场数：%d\n",sum);

return 0;

}

运行结果:



算法实验11问题：还是通常工程

代码：

#include <iostream>

#include <stdio.h>

#include <algorithm>

using namespace std;

int tree[101];

struct Edge{

int a;

int b;

int value;

}edges[6000];

int findRoot(int a){

if(tree[a]==-1) return a;

else{

int tmp = findRoot(tree[a]);

tree[a] = tmp;

return tmp;

}

}

bool cmp(Edge a,Edge b){

return a.value<b.value;

}

int main(){

int n;

while (scanf("%d",&n)!=EOF&&n!=0){

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

cin>>edges[i].a>>edges[i].b>>edges[i].value;

}

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

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

tree[j]=-1;

}

int ans=0;

for (int k = 1; k < n\*(n-1)/2; ++k) {

int a,b;

a = findRoot(edges[k].a);

b = findRoot(edges[k].b);

if(a!=b){

tree[a]=b;

ans+=edges[k].value;

}

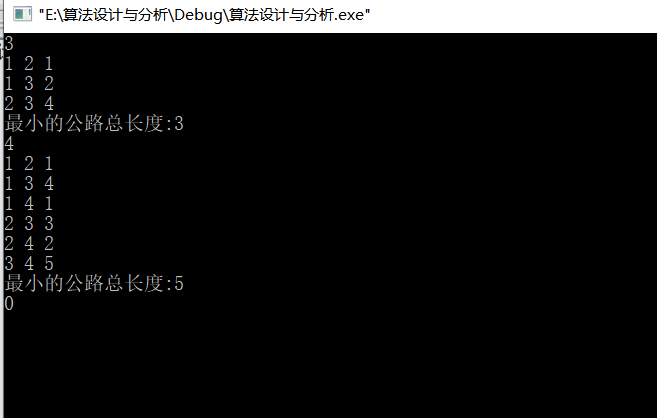
}

cout<<"最小的公路总长度:"<<ans<<endl;

}

}

代码运行结果：



算法实验12问题：捡苹果

代码：C++

#include <iostream>

#include <algorithm>

#include <cstring>

using namespace std;

const int inf = 0x3f3f3f3f;

typedef long long ll;

struct Node {

ll s,p;

double v;

}node[4];

bool cmp(Node p,Node q) {

return p.v < q.v;

}

ll dp[1101];

int max(int a,int b)

{

if(a>b)

return a;

else

return b;

}

int main()

{

int i,t,v,k = 0;

cin >> t;

while(t--) {

k++;

for(i = 0; i < 3; i++) {

cin >> node[i].s >> node[i].p;

node[i].v = node[i].s\*1.0/node[i].p;

}

cin >> v;

sort(node,node + 3,cmp);

ll ans = 0;

if(v > 1000) {

ans = ((v-1000)/node[0].s)\*node[0].p;

v = 1000 + (v-1000)%node[0].s;

}

memset(dp,0,sizeof(dp));

for( i = 0; i < 3; i++) {

for(ll j = node[i].s; j <= v; j++) {

dp[j] = max(dp[j],dp[j-node[i].s] + node[i].p);

}

}

ans += dp[v];

cout <<"Case " << k << ": " << ans << endl;

}

return 0;

}

运行结果：

