田忌赛马

时间限制：C/C++语言 1000MS；其他语言 3000MS

内存限制：C/C++语言 65536KB；其他语言 589824KB

题目描述：

田忌和齐王赛马，两人各有n匹马，输入第1行是田忌的赛马速度，第2行是齐王的，分别是两个无序正整数组给出，数字越大代表越快。齐王的赛马出场顺序总是从快到慢，问:田忌合理安排顺序后，最多能赢几场比赛?(注意：速度相同不算赢)

输入

两行，每行为一个数组

输出

数字表示田忌能赢得场数

样例输入

3 1 5

6 2 4

样例输出

2

#include<iostream>

#include<sstream>

#include<algorithm>

#include<vector>

using namespace std;

bool cmp(const int& a,const int& b)

{

return a>b;

}

int main()

{

string s1,s2;

char c;

int k;

getline(cin,s1);

getline(cin,s2);

vector<int> v1,v2;

stringstream ss(s1);

while(ss)

{

ss>>k;

v1.push\_back(k);

}

stringstream sss(s2);

while(sss)

{

sss>>k;

v2.push\_back(k);

}

v1.pop\_back();

v2.pop\_back();

int n=v1.size();

sort(v1.begin(),v1.end(),cmp);

sort(v2.begin(),v2.end(),cmp);

int ans=0;

for(int i=0,j=0;j<n;)

{

if(v1[i]>v2[j])

{

ans++;

i++;

}

j++;

}

cout<<ans<<endl;

return 0;

}

第2题：

二叉树路径和

**时间限制：**C/C++语言 1000MS；其他语言 3000MS  
**内存限制：**C/C++语言 65536KB；其他语言 589824KB

**题目描述：**

给定二叉树，每个节点value是非负整数，问:有多少条从根节点出发和等于K的路径。

输入

第1行为数字和K

之后输入的每个数字代表节点按层遍历的Value，N代表节点不存在。

例，输入 1 5 11 3 2 N 5 3 2 N 1 10 代表的二叉树如下。其中第3层只存在3个节点，故第4层最多出现8个数字或N。在结尾的N可能会省略。

      1

     /  \

    5   11

   / \      \

  3   2     5

 / \    \    /

3  2   1  10

输出

一个整数，表示路径数

样例输入

12

1 5 11 3 2 N 5 3 2 N 1 10

样例输出

2

#include<iostream>

#include<vector>

#include<algorithm>

#include<sstream>

#include<stdio.h>

using namespace std;

int ans;

int dfs(vector<int> v,int i,int n,int tmp,int k)

{

if(tmp==k)

{

ans++;

}

int l=2\*i+1;

int r=2\*i+2;

if(l<n&&v[l]!=-1)

{

dfs(v,l,n,tmp+v[l],k);

}

if(r<n&&v[r]!=-1)

{

dfs(v,r,n,tmp+v[r],k);

}

return 0;

}

int main()

{

int k;

cin>>k;

string s;

vector<int> v;

getchar();

getline(cin,s);

stringstream ss(s);

while(ss)

{

string t;

ss>>t;

if(t=="N")

{

v.push\_back(-1);

}

else

{

stringstream ts(t);

long long tt;

ts>>tt;

v.push\_back(tt);

}

}

v.pop\_back();

for(int i=0;i<v.size();i++)

{

if(v[i]==-1)

{

if(i\*2<v.size()-1)

{

v.insert(v.begin()+i\*2+1,-1);

v.insert(v.begin()+i\*2+1,-1);

}

}

}

int n=v.size();

ans=0;

if(n==0)

{

cout<<0<<endl;

return 0;

}

dfs(v,0,n,v[0],k);

cout<<ans<<endl;

return 0;

}

旅行商问题

**时间限制：**C/C++语言 1000MS；其他语言 3000MS  
**内存限制：**C/C++语言 65536KB；其他语言 589824KB

**题目描述：**

小Q想以最短路径环游全国n个城市，他发现这个著名的旅行商问题是NP-hard的，故退而寻找一个简化版本为:限制路线规则为从最左端城市出发，严格只向右走直至最右端城市(此时可能并不遍历每个城市)，再严格只向左走直至回到最左出发点(此时完成遍历所有城市)。已知每个点的x坐标不等，两个城市间的距离即为其二维平面上的直线距离，这种情况下，请写程序输出最小距离，保留两位小数。

输入

输入的第1行，为城市数n，之后n行，分别为n个城市的x、y轴坐标。

输出

输出最小距离，保留两位小数。

样例输入

4

1 1

2 3

3 1

4 2

样例输出

7.89

第3题：

#include <cmath>

#include <cstdio>

#include <iostream>

#include <cstring>

#include <algorithm>

using namespace std;

const int mm=1111;

struct data

{

double x,y;

}g[mm];

double d[mm][mm],f[mm][mm];

int i,j,k,n;

bool cmp(const data &a,const data &b)

{

return a.x<b.x;

}

double mdis(const data &a, const data &b)

{

return sqrt((a.x-b.x)\*(a.x-b.x)+(a.y-b.y)\*(a.y-b.y));

}

int main()

{

scanf("%d",&n);

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

scanf("%lf%lf",&g[i].x,&g[i].y);

sort(g,g+n,cmp);

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

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

{

d[i][j]=mdis(g[i],g[j]);

f[i][j]=1e30;

}

f[0][1]=d[0][1];

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

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

{

f[i][j+1]=min(f[i][j+1],f[i][j]+d[j][j+1]);

f[j][j+1]=min(f[j][j+1],f[i][j]+d[i][j+1]);

}

double ans=1e30;

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

ans=min(ans,f[j][n-1]+d[j][n-1]);

printf("%.2lf\n",ans);

return 0;

}