

[报告]poj 1789 Truck History

[Source]

<http://poj.org/problem?id=1789>

[Description]

给出 n 个长度相同的字符串，定义了一个距离函数可以计算每两个字符串之间的距离，已知一开始只有其中的一个字符串，其他的 $n-1$ 个字符串都是通过这个字符串或者是通过已派生出来的字符串派生出来的，每个字符串只由一个字符串派生出来，花费为两个字符串之间的距离，求所有可能的派生关系中，总花费最小的。

[Solution]

由题意可知，如果把派生关系看作一条无向边，把每个字符串看做一个点，则最后得出的图是一棵树。所以所求的实际上就是一个完全图的最小生成树。

[Code]

```
#include<iostream>
#include<stdio.h>
#include<string.h>
using namespace std;
const int N = 2100;
const int len=7;
const int inf = 1000;
int dis[N];
int d[N][N];
bool vis[N];
char s[N][10];
inline int cal(char a[],char b[])
{
    int ans=0;
    for(int i=0;i<len;i++)
        if(a[i]!=b[i])
            ans++;
    return ans;
}
int prim(int n)
{
    int i,j,k,a,b,mm,tmp,ans=0;

    for(i=2;i<=n;i++)
```

```

        dis[i]=d[1][i];
memset(vis,0,sizeof(vis));
vis[1]=1;
for(k=2;k<=n;k++){
    for(i=1,mm=-1;i<=n;i++){

        if(vis[i]==0&&(mm==-1||dis[i]<dis[mm]))
            mm=i;
    }
    vis[mm]=1;
    ans+=dis[mm];
    for(i=1;i<=n;i++)
        dis[i]=min(dis[i],d[mm][i]);
}
return ans;
}
void init(int n)
{
    int i,j,k;

    for(i=1;i<=n;i++)
        scanf("%s",s[i]);
    for(i=1;i<=n;i++)
        d[i][i]=0;
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            d[i][j]=d[j][i]=cal(s[i],s[j]);
}
int main()
{
    int n;

    while(scanf("%d",&n)==1&& n){
        init(n);
        cout<<"The highest possible quality is
1/"<<prim(n)<<'. '<<endl;
    }
    return 0;
}

```