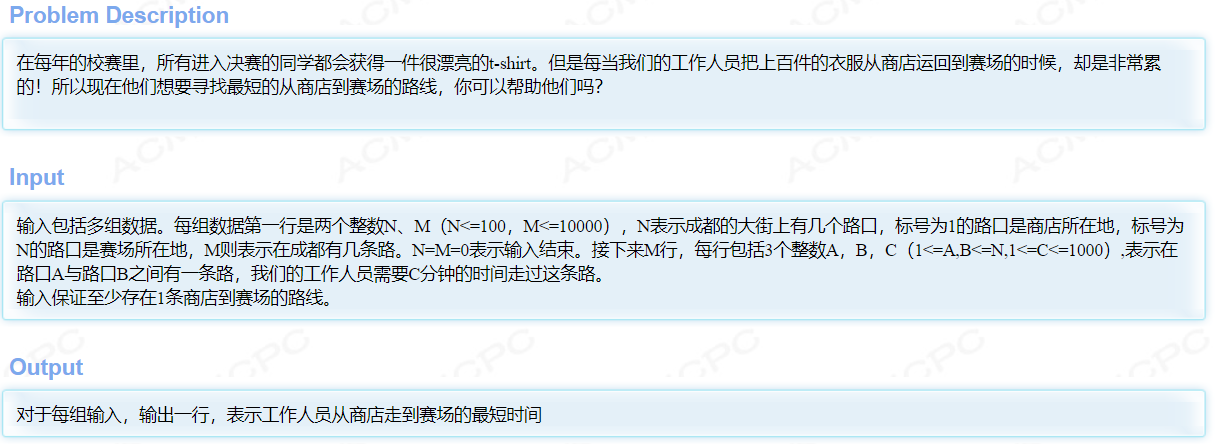
**Dijkstra算法求最短路**

因为整个实验重点都在搜索算法，所以这一次我决定把Dijkstra算法求最短路写一写，加深一下我原来在数据结构课上学习的这个的理解，其实这个的思想也很简单，就是不断地使用贪心策略就好了。

因为这个也是acm竞赛中一个常考地知识点，在各个高校地oj上关于这个算法地题目就非常多，所以这次我去杭电oj上面选择了一道关于这个算法的题目作为这次的实验内容，hdu2544

**题目：**



**思路：**直接把Dijkstra算法跑一遍就好了，没有什么其他的花里胡哨的思维障碍。

**Ac代码：**

#include<stdio.h>

#include<string.h>

#define INF 0x3f3f3f3f

int mp[1010][1010],dis[1010],book[1010],n,m;

void dijkstra()

{

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

{

dis[i]=mp[1][i];

}

for(int i=1;i<=n-1;i++)//找n-1轮

{

int minn=INF,u;

for(int j=1;j<=n;j++)//找与上一个点最近的点

{

if(book[j]==0&&dis[j]<minn)

{

minn=dis[j];

u=j;

}

}

book[u]=1;

for(int j=1;j<=n;j++)//找着最近的点后通过这个最近的点更新其余的点到起点的距离

{

if(book[j]==0&&dis[u]+mp[u][j]<dis[j])//book[j]==0和dis[j]<INF看心情写吧= =

{

dis[j]=dis[u]+mp[u][j];

}

}

}

printf("%d\n",dis[n]);

}

int main()

{

while(~scanf("%d%d",&n,&m)&&n+m)

{

for(int i=1;i<=n;i++)//下标从0还是1开始由题目的标号决定

{

for(int j=1;j<=n;j++)//如果是0建议都从0开始计数,是1则1

{

if(i==j) mp[i][j]=0;

else mp[i][j]=INF;

}

}

memset(book,0,sizeof(book));

int u,v,w;

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

{

scanf("%d%d%d",&u,&v,&w);

if(w<mp[u][v]) mp[u][v]=mp[v][u]=w;

}

dijkstra();//这道题固定起点为1,也可在函数里自定义起点

}

return 0;

}

**Ac截图：**

