

奶牛的旅行

一. 考察内容:

01分数规划 最短路径

二. 题目分析:

[题目大意]

在一张有向图上选择一个起点，找到一个点权和除以边权和最大的环，其中一个点经过多次只计算一次。

[写题思路]

经典01分数规划问题，首先需要证明，答案路径一定是一个简单环（哈密顿回路），因为如果存在一个点被经过多次的情况，则该环可以拆出两个子环，选择只走其中更大的环的方案不会比原方案差。

二分一个答案ans，使得 $\sum V_V / \sum V_E \geq ans$ ，则

$\sum V_V \geq \sum V_E * ans \rightarrow \sum V_V - \sum (V_E * ans) \geq 0$ ，最终输出结果即可。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 15
*文件类型: 源代码文件
*题目来源: BZOJ
*当前状态: 已通过
*备忘录: 最短路径 01分数规划
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 1001
#define MAXM 5001
int n,m,head[MAXN],_edge,v[MAXN],num[MAXN];
double d[MAXN],nowans;
struct EDGE
{
    int a,b,next;
    double v;
    EDGE(int a=0,int b=0,int v=0,int next=0):a(a),b(b),v(v),next(next){}
}edge[MAXM];
/* Variable explain:

*/
void adde(int a,int b,int v)
{
    edge[++_edge]=EDGE(a,b,v,head[a]);
    head[a]=_edge;
}
void read()
{

```

```

int ls1,ls2,ls3;
scanf("%d%d",&n,&m);
for(int i=1;i<=n;++i)scanf("%d",&v[i]);
for(int i=1;i<=m;++i)scanf("%d%d%d",&ls1,&ls2,&ls3),adde(ls1,ls2,ls3);
return;
}
bool spfa()
{
    deque<int> q;
    bitset<MAXN> vis;
    for(int i=1;i<=n;++i)d[i]=num[i]=0,vis[i]=1,q.push_back(i);
    while(!q.empty())
    {
        int a=q.front();
        q.pop_front();
        vis[a]=false;
        for(int i=head[a];i;i=edge[i].next)
        {
            int b=edge[i].b;
            if(d[b]<d[a]+v[b]-edge[i].v*nowans)
            {
                d[b]=d[a]+v[b]-edge[i].v*nowans;
                if(!vis[b])q.push_back(b),++num[b],vis[b]=true;
                if(num[b]>n)return true;
            }
        }
    }
    return false;
}
void solve()
{
    double l=0,r=1000;
    for(int i=0;i<20;++i)
    {
        nowans=(l+r)/2;
        if(spfa())l=nowans;
        else r=nowans;
    }
    printf("%.2lf",l);
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    solve();
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)