信奥算法

最小生成树问题

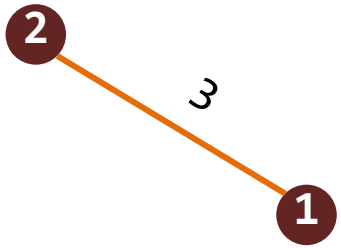
Kruskal算法

Minimum Spanning Tree

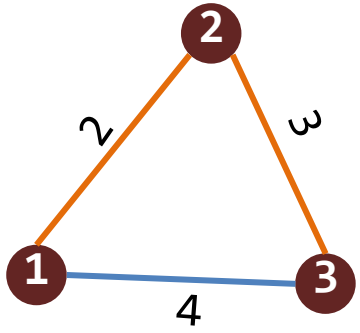
建桥

你拥有n个小岛，编号1到n。作为岛主，你找人设计了m座桥，第i座桥连接ai号和bi号岛，造价为wi。你希望所有岛之间都能通过桥直接或间接连通，请问至少要花多少钱？
保证能找到连通方案。输入n,m和m座桥信息，n<=100,m<=500

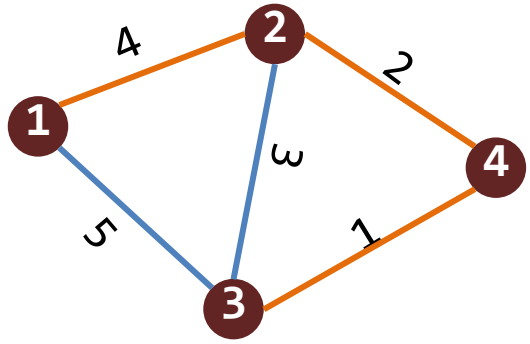
输入样例	输出样例
2 1 1 2 3	3



输入样例	输出样例
3 3 1 2 2 2 3 3 3 1 4	5



输入样例	输出样例
4 5 1 2 4 1 3 5 2 3 3 2 4 2 3 4 1	7



最小生成树MST

在无向图中，任意两个顶点都有路径相通，称为**连通图**

连通图的**生成树**是包含原图n个顶点和n-1条边的一棵树

最小生成树 的所有边的长度总和是生成树里最小的

n个顶点的生成树有n-1条边，若再添加一条边，**必定成环**

Kruskal算法

贪心 每次找最短边，*尝试*加入最小生成树

根据边的长度从小到大排序 ← O(mlogm)

依次查看每条边k：端点u_k,v_k,边长w_k

若u_k和v_k间还没有连通

将这条边k选入最小生成树答案

MST总长度增加w_k

否则，忽略这条边k

如何判定 **并查集**

```
5  struct edge{int u,v,w;};
6  edge e[M];
7  int n,m,id[N];
8  int root(int u){
9      return id[u]==u?u:id[u]=root(id[u]);
10 }
11 bool cmp(const edge&a,const edge&b){
12     return a.w<b.w;
13 }
```

边集数组:e[k]代表k号边

root(u)找u的老祖宗

排序规则:边权越小越靠前

```
27 int main(){
28     cin>>n>>m;
29     for(int i=0;i<m;i++)
30         cin>>e[i].u>>e[i].v>>e[i].w;
31     Kruskal();
32     return 0;
33 }
```

输入边集数组

```
14 void Kruskal(){
15     sort(e,e+m,cmp);
16     for(int u=1;u<=n;u++)id[u]=u;
17     int ans=0;
18     for(int k=0;k<m;k++){
19         int ru=root(e[k].u);
20         int rv=root(e[k].v);
21         if(ru==rv)continue;
22         id[ru]=rv;
23         ans+=e[k].w;
24     }
25     cout<<ans<<endl;
26 }
```

根据边的长度从小到大排序

并查集初始化

依次查看每一条边

若该边端点已连通
忽略这条边

将该边选入MST

总长度增加w_k

时间复杂度	O(mlogm)

高频错误

```
14 void Kruskal(){
15     sort(e,e+m,cmp);
16     for(int u=1;u<=n;u++)id[u]=u;
17     int ans=0;
18     for(int k=0;k<m;k++){
19         int ru=root(e[k].u);
20         int rv=root(e[k].v);
21         if(ru==rv)continue;
22         id[ru]=rv;
23         ans+=e[k].w;
24     }
25     cout<<ans<<endl;
26 }
```

记住对边排序

记住并查集初始化

总和很大时需要ll

记住先找老祖宗

已经连接的情况不能重复累加

Kruskal算法

贪心	每次找最短边，*尝试*加入最小生成树

请思考如何证明Kruskal算法的正确性

贪心法常用证明方法	反证法	✚	局部调整

证明： Kruskal算法 能找到MST

Kruskal选的第1条边e_1一定在某棵MST中

Kruskal选的前2条边e_1,e_2一定在某棵MST中

Kruskal选的前3条边e_1, e_2, e_3一定在某棵MST中

…… ……

Kruskal选的前n-1条边一定在某棵MST中

证明： Kruskal算法 能找到MST

Kruskal选的第1条边e_1一定在某棵MST中

反证法

假设存在1棵不包含e_1的MST记作T

局部调整

向T中**添加e_1，必定成环**

环中必有边长不小于e_1的边f，**删除f**

新的生成树**$T+e_1-f$**的边长总和不超过T

Kruskal选的前2条边e_1,e_2一定在某棵MST中

Kruskal选的前3条边e_1, e_2, e_3一定在某棵MST中

…… ……

Kruskal选的前n-1条边一定在某棵MST中

Kruskal算法

贪心	每次找最短边，*尝试*加入最小生成树

提前结束	如何加快运行速度？
	若已经找到MST的n-1条边 就可以终止算法

提前结束

```
14 void Kruskal(){
15     sort(e,e+m,cmp);
16     for(int u=1;u<=n;u++)id[u]=u;
17     int ans=0,cnt=0;
18     for(int k=0;k<m;k++){
19         int ru=root(e[k].u);
20         int rv=root(e[k].v);
21         if(ru==rv)continue;
22         id[ru]=rv;
23         ans+=e[k].w;
24
25         if(cnt==n-1)break;
26     }
27     cout<<ans<<endl;
28 }
```

cnt记录当前已经找到MST里的几条边

讨论

无向边要改成双向边吗?

重边如何处理?

最大生成树问题

请写出该问题的描述
已知什么求什么

已知n个节点的连通图里
m条边的每条边长
求一个生成树
要求树的n-1条边长总和最大

请用一句话描述算法步骤

每次找最长的边，尝试加入生成树

图论建模

将原问题描述转换为
抽象的图论问题描述

现场挑战
591

图论建模

输入
3
ABCDE
ABXYZ
ABDEX

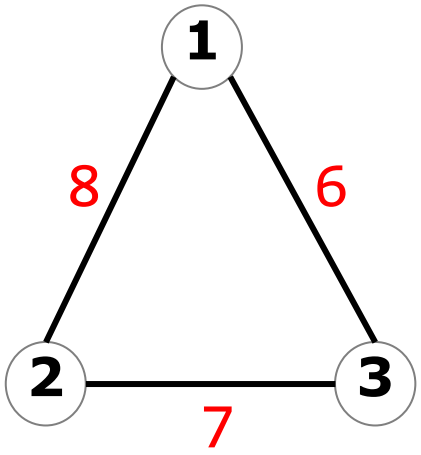
把原问题已知条件转为图论信息

每个节点代表什么含义	
每条边代表什么含义	
每条边长如何计算	
共有几条边?	n*(n-1)/2

把求解的问题转为图论的问题描述

图论建模

输入
3
ABCDE
ABXYZ
ABDEX



每个节点代表1个物种

每条边代表2个物种转换所消耗的能量
这2个物种基因序列中不同字符的个数

1号物种ABCDE和2号物种ABXYZ
基因序列里共8个不同字符

1号物种ABCDE和3号物种ABDEX
基因序列里共6个不同字符

2号物种ABXYZ和3号物种ABDEX
基因序列里共7个不同字符

求：该图的最小生成树

gene[i]表示
i号物种的基因序列

```
38      cin>>n;
39      for(int i=1;i<=n;i++)cin>>gene[i];
40      for(int i=1;i<=n-1;i++)
41          for(int j=i+1;j<=n;j++){
42              e[m].u=i;
43              e[m].v=j;
44              e[m].w=dst(i,j);
45              m++;
46          }
47      Kruskal();
```

dst(i,j)计算
i号点和j号点
之间的边长

m表示什么含义?

dst(i,j)计算
i号点和j号点
之间的边长

```
12 int dst(int i,int j){
13     set<char> s;
14     for(int k=0;k<5;k++){
15         s.insert(gene[i][k]);
16         s.insert(gene[j][k]);
17     }
18     return s.size();
19 }
```

gene[i]字符串
gene[j]字符串
这两条字符串里共有
多少种不同的字符

MST演示

算法可视化网址：
visualgo.net/en/mst

A **Spanning Tree (ST)** of a connected undirected weighted graph **G** is a subgraph of **G** that is a **tree** and **connects (spans) all vertices**.

A **Minimum Spanning Tree (MST)** of **G** is an ST of **G** that has the **smallest total weight** among the various STs.

自编题

仿照课堂例题，请自编一道编程题，要求以“**最小生成树**”的算法为核心求解步骤。鼓励加入各类算法元素，构成原问题的变种形式。本作业题的提交方式为：一个word文档发到课程微信群

word文档中需要提供：

1. 题目描述
2. 输入数据的范围
3. 输入格式
4. 输出格式
5. 输入样例至少1组
6. 输出样例至少1组
7. 标准答案程序1份

太戈编程

	1776
	591
	自编题
拓展题	592