

信奥算法



现场挑战 快快编程1942

快快编程
kkcoding.net

算法分析

离线询问	可以看完所有操作预先计算
------	--------------

识别特性	无环，不保证连通，森林结构
------	---------------

完成所有链接后所有路径距离均确定
针对 $lca(u, v)$, $dst(u, v)$ 问题
做好预计算准备工作

$tI[u], tO[u]$: 节点 u 在欧拉序列中的时间戳

$d[u]$: 根到节点 u 的路径长度

$p[u][i]$: 节点 u 的第 2^i 代祖先

并查集	顺序处理操作时动态维护连通性
-----	----------------

```
3 const int N=500009;  
4 struct Edge{int v,w,nxt;} e[N*2];  
5 struct Query{int t,u,v;} q[N];
```

```
17 scanf("%d %d",&n,&nQ);  
18 for(int i=1;i<=nQ;++i){  
19     scanf("%d %d",&x,&y);  
20     char ch=getchar();  
21     if(ch!=' '){  
22         q[i]=(Query){2,x,y};  
23         continue;  
24     }
```

判断当前行
是2个数还是3个数

```
25     scanf("%d",&w);  
26     add(x,y,w);  
27     add(y,x,w);  
28     q[i]=(Query){3,x,y};  
29 }
```

```
63 L=log2(n)+1;
64 for(int u=1;u<=n;++u)if(!vst[u])dfs(u,0);

65 for(int u=1;u<=n;++u)id[u]=u;
66 for(int i=1;i<=nQ;++i){
67     int u=q[i].u;
68     int v=q[i].v;
69     int ru=root(u);
70     int rv=root(v);
71     if(q[i].t==3){
72         id[ru]=rv;
73         continue;
74     }
75     if(ru!=rv)printf("-1\n");
76     else printf("%d\n",dst(u,v));
77 }
```

```

31 void dfs(int u,int fa){
32     vst[u]=1;
33     [ ]
34     p[u][0]=fa;
35     for(int i=1;i<=L;++i)
36         p[u][i]=[ ]
37     for(int i=hd[u];i;i=e[i].nxt){
38         int v=e[i].v;
39         if(v==fa)continue;
40         [ ]
41         dfs(v,u);
42     }
43     [ ]
44 }

```

有权边
 对应深度计算
 $d[v]$ 表示 v 到根
 的路径距离



现场挑战 快快编程1943

快快编程
kkcoding.net



为什么此题需要 高精度大数记录答案

一起学习高精度大数基本功

快快编程
kkcoding.net



请预测输出结果

```
1 #include<iostream>
2 #include<iomanip>
3 using namespace std;
4 typedef long long ll;
5 int main(){
6     ll x=12;
7     ll y=123;
8     ll z=123456789;
9     cout<<setw(5)<<setfill('0')<<x<<endl;
10    cout<<setw(5)<<setfill('0')<<y<<endl;
11    cout<<setw(5)<<setfill('0')<<z<<endl;
12    return 0;
13 }
```

```
4 const int SZ=1009;  
5 typedef int BigNum[SZ]; ←————
```

```
10 void convert(int *a,int s){
```

```
15 void add(int *a,int *b){
```

```
22 void mul(int *a, int b) {
```

```
28 void copy(int *a, int *b) {
```

```
31 bool eq0(int *a) {
```

```
35 void print(int *a){
```



```
174 cin>>n>>m>>k;  
175 for(int i=1;i<=m;++i)cin>>A[i]>>B[i];  
176 if(n==1) solve1();  
177 else if(m==0&& n<=9) solveSmallNum();  
178 else if(m==0) solveBigNum();  
179 else if(n<=9) solveSmallNumAB();  
180 else solveBigNumAB();
```



n=1时

```
40 void solve1(){  
41     cout<<k<<endl<<(1+k)*k/2<<endl;  
42 }
```

m=0时



方案数

$$cnt = k^n$$

个位总和

$$sum_0 = 1 \times k^{n-1} + 2 \times k^{n-1} + \dots + k \times k^{n-1} = \frac{k(k+1)}{2} k^{n-1}$$

十位总和

$$sum_1 = \frac{k(k+1)}{2} k^{n-1} \times 10$$

第n位总和

$$sum_{n-1} = \frac{k(k+1)}{2} k^{n-1} \times 10^{n-1}$$

$$sum = sum_0 + sum_1 + \dots + sum_{n-1}$$

m=0时



```
43 void solveSmallNum(){
44     ll a=1;
45     for(ll i=1;i<=n-1;++i)a*=k;
46     ll cnt=a;
47     cnt*=k;
48     ll b=k*(k+1)/2;
49     a*=b;
50     ll sum=0;
51     for(ll i=0;i<n;++i){
52         sum+=a;
53         a*=10;
54     }
55     cout<<cnt<<endl;
56     cout<<sum<<endl;
57 }
```

$$a = k^{n-1}$$

$$k^n$$

$$a = \frac{k(k+1)}{2} k^{n-1}$$

m=0时

```
4 const int SZ=1009;
5 typedef int BigNum[SZ];
```



```
58 void solveBigNum(){
59     BigNum a,cnt,sum;
60     convert(a,1);
61     for(int i=1;i<=n-1;++i)mul(a,k);
62     copy(cnt,a);
63     mul(cnt,k);
64     int b=k*(k+1)/2;
65     mul(a,b);
66     convert(sum,0);
67     for(ll i=0;i<n;++i){
68         add(sum,a);
69         mul(a,10);
70     }
71     print(cnt);
72     print(sum);
73 }
```

$$a = k^{n-1}$$

$$k^n$$

$$a = \frac{k(k+1)}{2} k^{n-1}$$



每条限制都可以理解为左边已存在元素
对右边元素选择的约束

动态规划定义状态

$\text{cnt}[i][p]$ 表示从左到右 i 位
恰用了 p 二进制对应集合的数时
共几种合法方案

$\text{sum}[i][p]$ 表示从左到右 i 位
恰用了 p 二进制对应集合的数时
所有合法数总和

$p = (1011)_2$ 时表示恰用了 124

m>0时



```
74  /*
75  cnt[i][p]表示从左到右i位
76  恰用了p二进制对应集合的数时
77  共几种合法方案
78  sum[i][p]表示从左到右i位
79  恰用了p二进制对应集合的数时
80  所有合法数总和
81  p=(1011)2时表示恰用了124
82  n=2,m=1,k=2
83  A[1]=1,B[1]=2
84  cnt  p=00,01,10,11
85  i=0    1    0    0    0
86  i=1    0    1    1    0
87  i=2    

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|


88
89  sum  p=00,01,10,11
90  i=0    0    0    0    0
91  i=1    0    1    2    0
92  i=2    

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|


```

$m > 0$ 时



$\text{cnt}[i][p]$ 表示从左到右 i 位恰用了 p 二进制
对应集合的数时共几种合法方案

$\text{sum}[i][p]$ 表示从左到右 i 位恰用了 p 二进制
对应集合的数时所有合法数总和

```
int nPtn=(1<<k);
```

预计算 a 的左侧违规集合 $X[a]$

```
for(int i=1;i<=m;++i)  
    X[B[i]]|=(1<<A[i]-1);
```

$m > 0$ 时



$\text{cnt}[i][p]$ 表示从左到右 i 位恰用了 p 二进制
对应集合的数时共几种合法方案

$\text{sum}[i][p]$ 表示从左到右 i 位恰用了 p 二进制
对应集合的数时所有合法数总和

决策

推动法 push

枚举 a : 第 $i+1$ 个位置放哪个数

$\text{cnt}[i+1][p|(1 \ll a-1)] +=$

$\text{cnt}[i][p];$

$\text{sum}[i+1][p|(1 \ll a-1)] +=$

$\text{sum}[i][p]*10 + a*\text{cnt}[i][p];$

合法性

第 $i+1$ 个位置放 a 不可以和 p 冲突

$p \& X[a] == 0$ 表示没有违规



```
119 ll cnt[N][512],sum[N][512];
120 void solveSmallNumAB(){
121     for(int i=1;i<=m;++i)X[B[i]]=(1<<A[i]-1);
122     int nPtn=(1<<k);
123     cnt[0][0]=1;
124     for(int i=0;i<=n-1;++i)
125         for(int p=0;p<nPtn;++p){
126             if(cnt[i][p]==0)continue;
127             for(int a=1;a<=k;++a){
128                 if(p&X[a])continue;
129                 cnt[i+1][p|(1<<a-1)]+=cnt[i][p];
130                 sum[i+1][p|(1<<a-1)]+=sum[i][p]*10+a*cnt[i][p];
131             }
132         }
133     ll C=0,S=0;
134     for(int p=0;p<nPtn;++p){
135         C+=cnt[n][p];
136         S+=sum[n][p];
137     }
138     cout<<C<<endl;
139     cout<<S<<endl;
140 }
```

m>0时

大数据

80分TE

```
141 BigNum c[2][512],s[2][512],C,S;
142 void solveBigNumAB(){
143     for(int i=1;i<=m;++i)X[B[i]]|=(1<<A[i]-1);
144     int nPtn=(1<<k);
145     convert(c[0][0],1);
146     for(int i=0;i<=n-1;++i){
147         for(int p=0;p<nPtn;++p)
148             convert(c[i&1^1][p],0), convert(s[i&1^1][p],0);
149         for(int p=0;p<nPtn;++p){
150             if(eq0(c[i&1][p]))continue;
151             mul(s[i&1][p],10);
152             for(int a=1;a<=k;++a){
153                 add(s[i&1][p],c[i&1][p]);
154                 if(p&X[a])continue;
155                 add(c[i&1^1][p|(1<<a-1)],c[i&1][p]);
156                 add(s[i&1^1][p|(1<<a-1)],s[i&1][p]);
157             }
158         }
159     }
160     convert(C,0); convert(S,0);
161     for(int p=0;p<nPtn;++p)
162         add(C,c[n&1][p]), add(S,s[n&1][p]);
163     print(C); print(S);
164 }
```



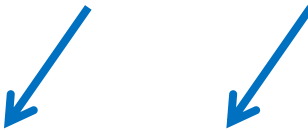
加速思路
高精度数组每个元素放**10**位数字
而不是**1**位数字

```
3 typedef long long ll;  
4 const ll nDigits=10;  
5 const ll U=1e10;  
6 const ll SZ=109;  
7 typedef ll BigNum[SZ];
```

对应修改程序



```
30 void convert(ll *a, ll s){  
31     ll i;  
32     for(i=0; i<SZ&& s; i++) a[i]=s%U, s/=U;  
33     for(; i<SZ; i++) a[i]=0;  
34 }
```



对应修改程序



```
55 void print(ll *a){  
56     ll i;  
57     for(i=SZ-1;i>0;i--) if(a[i]) break;  
58     cout<<a[i--];  
59     for(;i>=0;i--)  
60         cout<<setw(nDigits)<<setfill('0')<<a[i];  
61     cout<<endl;  
62 }
```

对应修改程序

快快编程作业

1942

1943