

# Code Templates

Seaside Guys

2021 年 12 月 4 日

## 目录

1	倍增 lca	1
2	数位 dp	2
3	线性欧拉函数	3

## 1 倍增 lca

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #define maxx 500002
5 using namespace std;
6 int head[maxx],d[maxx],p[maxx][21];
7 int k=0,n,m,s,x,y;
8 struct node
9 {
10     int v,next;
11 } e[maxx*2];
12 void add(int u,int v)
13 {
14     e[k].v=v;
15     e[k].next=head[u];
16     head[u]=k++;
17 }
18 void dfs(int u,int fa)
19 {
20     d[u]=d[fa]+1;
21     p[u][0]=fa;
22     for(int i=1;(1<<i)<=d[u];i++)
23         p[u][i]=p[p[u][i-1]][i-1];
24     for(int i=head[u];i!=-1;i=e[i].next)
25     {
26         int v=e[i].v;
27         if(v!=fa)
28             dfs(v,u);
29     }
30 }
31 int lca(int a,int b)
32 {
33     if(d[a]>d[b])
34         swap(a,b);
35     //int dc=d[a]-d[b];
36     /*for(int i=20;i>=0;i--)
37         if(dc&(1<<i))
38             a=p[a][i];
39     */
40     for(int i=20;i>=0;i--)
41     {
```

```

42         if(d[a]<=d[b]-(1<<i))
43             b=p[b][i];
44     }
45     if(a==b)
46         return a;
47     for(int i=20;i>=0;i--)
48     {
49         if(p[a][i]!=p[b][i])
50             a=p[a][i],b=p[b][i];
51     }
52     return p[a][0];
53 }
54 int main()
55 {
56     scanf("%d%d%d",&n,&m,&s);
57     memset(head,-1,sizeof(head));
58     for(int i=1;i<=n-1;i++)
59     {
60         scanf("%d%d",&x,&y);
61         add(x,y);
62         add(y,x);
63     }
64     dfs(s,0);
65     for(int i=1;i<=m;i++)
66     {
67         scanf("%d%d",&x,&y);
68         printf("%d\n",lca(x,y));
69     }
70     return 0;
71 }

```

## 2 数位 dp

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int dp[8][12][2],dight[12];
4 int n,m,cnt;
5 int dfs(int pos,int last,bool limit)
6 {
7     int ans=0;
8     if(pos==cnt)
9         return 1;

```

```

10     if(dp[pos][last][limit]!=-1)
11         return dp[pos][last][limit];
12     for(int i=0;i<= (limit==true ? dight[pos] : 9);i++)
13     {
14         if((i==4)||(i==2 && last==6))
15             continue;
16         ans+=dfs(pos+1,i,(limit==true)&&(i==dight[pos]));
17     }
18     dp[pos][last][limit]=ans;
19     return ans;
20 }
21 int solve(int x)
22 {
23     cnt=0;
24     memset(dp,-1,sizeof(dp));
25     memset(dight,0,sizeof(dight));
26     while(x)
27         dight[cnt++]=x%10,x/=10;
28     reverse(dight,dight+cnt);
29     return dfs(0,11,true);
30 }
31 }
32 int main()
33 {
34     while(1)
35     {
36         cin>>n>>m;
37         if(n==0&&m==0)
38             return 0;
39         cout<<solve(m)-solve(n-1)<<endl;
40     }
41     return 0;
42 }

```

### 3 线性欧拉函数

```

1 #include<bits/stdc++.h>
2 #define ll long long
3 #define maxn 100000010
4 using namespace std;
5 ll phi[maxn],n;
6 bool isnp[maxn];

```

```

7  vector<ll> primes;
8  void init(ll n)
9  {
10     phi[1]=1;
11     for(ll i=2;i<=n;i++)
12     {
13         if(!isnp[i])
14             primes.push_back(i),phi[i]=i-1;
15         for(ll p=0;p<primes.size();p++)
16         {
17             if(primes[p]*i>n) break;
18             isnp[primes[p]*i]=1;
19             if(i%primes[p]==0)
20             {
21                 phi[primes[p]*i]=phi[i]*primes[p];
22                 break;
23             }
24             else phi[primes[p]*i]=phi[primes[p]]*phi[i];
25         }
26     }
27 }
28 int main()
29 {
30     cin>>n;
31     init(n);
32     cout<<phi[n];
33     return 0;
34 }

```