牛客练习赛77

比赛链接

E-小G的GLS图

题意

给定 n 个点的点权图，u,v 之间有连边当且仅当 $\text{gcd}(u,v)\neq 1$。

问图中满足删去该点后图连通分量增加的点的数量。

题解

将每个点向该点的点权值的素因子连边，然后求这 n 个点中割的数量。

但需要注意当某个素因子的度仅为1时要提前删去该素因子，否则与该素因子连边的点一定会被记为割，但实际上该点不一定是原图的割。

总时间复杂度 $O(v+N\sqrt v)$。

```
const int MAXV=1e7+5,MAXN=1e6+5,MAXM=2e6+5;
int visp[MAXV],prime[MAXV],pid[MAXV],p_cnt;
void get_p(){
    _for(i,2,MAXV){
        if(!visp[i])prime[p_cnt]=i,pid[i]=p_cnt++;
        for(int j=0;j<p_cnt&&i*prime[j]<MAXV;j++){
            visp[i*prime[j]]=1;
            if(i%prime[j]==0)
            break;
        }
    }
}
struct Edge{
    int to,next;
}edge[MAXM];
int head[MAXN],edge_cnt;
void Insert(int u,int v){
    edge[++edge_cnt]=Edge{v,head[u]};
    head[u]=edge_cnt;
}
int a[MAXN],deg[MAXN];
int low[MAXN],dfs_id[MAXN],dfs_t;
bool iscut[MAXN],vis[MAXN];
void dfs(int u,int fa){
    low[u]=dfs_id[u]=++dfs_t;
```

```
    int child=0;
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        if(!dfs_id[v]){
            dfs(v,u);
            low[u]=min(low[u],low[v]);
            if(low[v]>=dfs_id[u]&&u!=fa)
            iscut[u]=true;
            child++;
        }
        else
        low[u]=min(low[u],dfs_id[v]);
    }
    if(u==fa&&child>=2)
    iscut[u]=true;
}
int main()
{
    get_p();
    int n=read_int();
    _for(i,0,n){
        a[i]=read_int();
        int t=a[i];
        for(int j=0;prime[j]*prime[j]<=t;j++){
            if(t%prime[j]==0){
                while(t%prime[j]==0)t/=prime[j];
                deg[j]++;
            }
        }
        if(t!=1)deg[pid[t]]++;
    }
    _for(i,0,n){
        int t=a[i];
        for(int j=0;prime[j]*prime[j]<=t;j++){
            if(t%prime[j]==0){
                while(t%prime[j]==0)t/=prime[j];
                if(deg[j]!=1){
                    Insert(i+p_cnt,j);
                    Insert(j,i+p_cnt);
                }
            }
        }
        if(t!=1&&deg[pid[t]]!=1){
            Insert(pid[t],i+p_cnt);
            Insert(i+p_cnt,pid[t]);
        }
    }
    _for(i,0,n){
        if(!dfs_id[i+p_cnt])
```

```
        dfs(i+p_cnt,i+p_cnt);
    }
    int ans=0;
    _for(i,0,n)ans+=iscut[i+p_cnt];
    enter(ans);
    return 0;
}
```