PE 355

一个很自然的想法是取出所有质数不大于n的最大的幂，~~然后我们发现它挂得飞起~~

跑一下$n=30$的数据我们发现，我们跑出来的最优解是

$Co(n)=\{1,16,27,25,7,11,13,17,19,23,29\}$

然而事实上，把16和7用28替换会更加的优秀

$Co(n)=\{1,27,25,11,13,17,19,23,28,29\}$

~~那怎么办啊我现在不会做了~~

接下来我们猜一波结论

> 最后这个集合中的任意一个数都不会有超过两个质因子
>
> 并且如果某个数含有两个质因子，那么其中必定有一个会$\geq\sqrt{n}$
>
> ~~听起来有理有据不过并不会证~~

那么我们先把$\frac{n}{2}$的质数都丢掉，然后把剩下的这些质数拿出来，$\geq\sqrt{n}$的放一边，$\leq\sqrt{n}$的放一边

剩下的就只要跑个最大带权匹配就好

```
#include<bits/stdc++.h>
#define read() Read<int>()
using namespace std;
typedef long long ll;
const int N=2e5+5;
ll nn=N-5,Ans,p[N];
int isprime[N],prime[N],ptot;
inline void get_prime(int prange){
    isprime[1]=1;
    for (int i=2;i<=prange;++i){
        if (!isprime[i]) prime[++ptot]=i;
        for (int j=1;j<=ptot && prime[j]*i<=prange;++j){
            isprime[i*prime[j]]=1;
            if (!(i%prime[j])) break;
        }
    }
}
namespace dinic{
    #define int ll
    const ll inf=1e18;
    struct edge{
        int next,to,w,c;
    }a[N<<1];
    int head[N],cnt;
    inline void add(int u,int v,int w,int c){
        a[++cnt].to=v;
        a[cnt].next=head[u];
```

```
    head[u]=cnt;
    a[cnt].w=w;
    a[cnt].c=c;
}
int n,m,S,T;
inline void input(){
    cnt=1;
    S=1;
    for (int i=1;i<=86;++i){
        add(S,i+1,1,0);
        add(i+1,S,0,0);
    }
    n=87;
    for (int i=87;i<=ptot;++i){
        if (prime[i]>100000) break;
        ++n;
        for (int j=1;j<=86;++j){
            ll x=prime[i];
            while (x*prime[j]<=nn) x*=prime[j];
            if (x<=p[i]+p[j]) continue;
            add(j+1,i+1,1,x-p[i]-p[j]);
            add(i+1,j+1,0,p[i]+p[j]-x);
        }
    }
    T=++n;
    for (int i=87;i<=ptot;++i){
        if (prime[i]>100000) break;
        add(i+1,T,1,0),add(T,i+1,0,0);
    }
}
queue<int>q;
int vis[N],d[N],pre[N],las[N],f[N];
inline bool spfa(){
    q=queue<int>();
    memset(vis,0,sizeof(vis));
    memset(pre,0,sizeof(pre));
    for (int i=1;i<=n;++i) d[i]=-inf;
    f[S]=inf,vis[S]=1,q.push(S),d[S]=0;
    while (!q.empty()){
        int x=q.front();
        q.pop(),vis[x]=0;
        for (int i=head[x];i;i=a[i].next){
            int y=a[i].to;
            if (d[y]<d[x]+a[i].c && a[i].w){
                d[y]=d[x]+a[i].c;
                pre[y]=x,las[y]=i;
                f[y]=min(f[x],a[i].w);
                if (!vis[y]) vis[y]=1,q.push(y);
            }
        }
    }
    return pre[T];
}
```

```
    inline ll solve(){
        input();
        ll ans1=0,ans2=0;
        while (spfa()){
            ans1+=f[T],ans2+=f[T]*d[T];
            for (int i=T;i;i=pre[i]){
                a[las[i]].w-=f[T];
                a[las[i]^1].w+=f[T];
            }
        }
        return ans2;
    }
}
signed main(){
    get_prime(nn);
    for (int i=1;i<=ptot;++i){
        ll x=prime[i];
        while (x*prime[i]<=nn) x*=prime[i];
        Ans+=(p[i]=x);
    }
    printf("%lld\n",Ans+dinic::solve()+1);
    return 0;
}
```