model pollard_rho

标准

```
#include<bits/stdc++.h>
 2
    using namespace std;
    namespace MR{
 5
         const int times=50;
         // 乘法防止溢出, 如果p * p不爆LL的话可以直接乘; O(1)乘法或者转化成二进制加法
 6
         long long qmul(long long x,long long y,long long mod){ return (x*y-(long bounds))
    long)(x/(long double)mod*y+le-3)*mod+mod)%mod; }
         long long qpow(long long a, long long b, long long p)\{a\%=p; long long\}
    ret=1; for(;b;b>>=1,a=qmul(a,a,p)) if (b\&1) ret=qmul(ret,a,p); return ret; }
 9
         bool Miller_Rabin(long long n){
10
             if(n<3) return n==2;</pre>
11
             long long u=0, v=n-1;
             while(v\%2==0) u++, v>>=1;
12
13
             for(int i=0;i<times;i++){</pre>
                 long long w=2+rand()\%(n-2), x=qpow(w,v,n);
14
15
                 if(x==1 | x==n-1) continue;
16
                 int j;
                 for(j=0; j<u; j++){}
17
18
                      x=qmul(x,x,n);
                      if(x==n-1) break;
19
21
                 if(j>=u) return 0;
22
23
             return 1;
         }
24
25
         long long f(long long x,long long c,long long mod){ return
    ((<u>__int128</u>)x*x+c)%mod; }
         long long find_factor(long long p){
26
             long long x,y,z,c=0,g; int i,j;
27
28
             while(1){
29
                 y=x=rand()%p;
30
                 z=1; c++;
31
                 i=0, j=1;
32
                 while(++i){
33
                      x=f(x,c,p);
34
                      z=(\underline{\hspace{1cm}}int128)z*abs(y-x)%p;
35
                      if(x==y||!z) break;
36
                      if(!(i%127)||i==j){
37
                          g=\underline{gcd}(z,p);
38
                          if(g>1) return g;
39
                          if(i==j) y=x, j<<=1;
                      }
40
41
                 }
42
             }
43
44
         void Pollard_Rho(vector<long long> &cnt,long long n){
             while(!(n\&1)) cnt.push_back(2), n>>=1;
45
             if(n==1) return;
46
47
             if(Miller_Rabin(n)) return cnt.push_back(n),void();
```

```
48
             long long p=find_factor(n);
49
             Pollard_Rho(cnt,n/p),Pollard_Rho(cnt,p);
50
        }
51
    }
52
    void solve(){
        srand((unsigned)time(NULL));
53
54
         long long n; cin>>n;
55
         vector<long long> res;
56
57
         MR::Pollard_Rho(res,n);
58
59
         map<long long,int> cnt;
60
         for(auto v:res) cnt[v]++;
         for(auto v:cnt)
61
62
             if(v.second>1){
                 cout<<"yes\n";</pre>
63
64
                 return;
65
         puts("no");
66
67
68
    int main(){
         int T; cin>>T;
69
70
         while(T--) solve();
71
    }
```

玄学

```
#include<bits/stdc++.h>
   1
  2
            using namespace std;
   3
   4
            namespace MR{
                         // 18位素数: 154590409516822759
   5
   6
                         // 19位素数: 2305843009213693951 (梅森素数)
   7
                         // 19位素数: 4384957924686954497
   8
                         long long prime[11] = {2,3,5,7,233,331,11,13,17,19,23};
  9
                         long long mi;
10
                         // 乘法防止溢出,如果p * p不爆LL的话可以直接乘; O(1)乘法或者转化成二进制加法
                         long long qmul(long long x,long long y,long long mod) { return (x*y-(long + long + l
11
             long)(x/(long double)mod*y+1e-3)*mod+mod)%mod; }
                         long long qpow(long long a, long long b, long long mod) { long long ret=1;
12
             for(;b;a=qmul(a,a,mod),b>>=1) if(b \le 1) ret=qmul(ret,a,mod); return ret; }
13
                         bool M_R(long long p){//传入值,返回0即为合数,犯为1即为质数,范围可测到<math>llong p)
14
                                     if(p==2) return 1;
15
                                      if(p<2||!(p&1)) return 0;
                                     long long s = p - 1;
16
17
                                     while(!(s&1)) s>>=1;
                                      for(int i=0;i<11;++i) {
18
19
                                                  if(p==prime[i]) return 1;
20
                                                  long long t=s,m=qpow(prime[i],s,p);
21
                                                 while(t!=p-1&&m!=1&&m!=p-1){
22
                                                               m=qmul(m,m,p);
23
                                                              t<<=1;
24
25
                                                  if(m!=p-1&&!(t&1)) return 0;
26
                                     }
```

```
27
            return 1;
28
        }
        long long f(long long x,long long mod,int a){ return
29
    ((__int128)x*x+a)%mod; }
30
        long long find_factorplus(long long N, long long seed){
31
            long long a=rand(),b=a,p;
32
            do{
                 a = f(a, N, seed);
33
34
                 b = f(f(b,N,seed),N,seed);
                 p = \underline{gcd}(abs(b - a), N);
35
                 if(p > 1\&\&p<N) return p;
36
37
            }while(b!=a);
38
             return 0;
39
        }
        void p_r(vector<long long> &cnt,long long x){
40
            while((x\&1)==0) cnt.push_back(2),x>>=1;
41
42
            if(x==1) return;
43
            if(M_R(x)) return cnt.push_back(x),void();
44
            long long p=0;
45
            while(p==0){
                long long seed=1+rand()%(x-1);
46
47
                p=find_factorplus(x,seed);
48
49
            p_r(cnt,p), p_r(cnt,x/p);
        }
50
51
52
    void solve(){
53
        srand((unsigned)time(NULL));
        long long n; cin>>n;
54
55
        vector<long long> res;
56
57
        MR::p_r(res,n);
58
        map<long long,int> cnt;
59
60
        for(auto v:res) cnt[v]++;
61
        for(auto v:cnt)
62
            if(v.second>1){
                 puts("yes");
63
64
                 return;
65
            }
66
        puts("no");
67
68
    int main(){
        int T; cin>>T;
69
        while(T--) solve();
70
71
    }
```