

原创 优化模板

2019-10-17 10:50:12 _Y-_Y- 阅读数 1 文章标签: ACM 更多

编辑

版权声明: 本文为博主原创文章, 遵循 CC 4.0 BY-SA 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44410512/article/details/102601351**01快速乘+龟速乘+快速幂+矩阵快速幂**

```
1  /*
2  a%b, 在整数除法下它等价于a - a/b * b
3
4  */
5  #define ll long long
6  inline ll mulit(ll x,ll y,ll mod)//01快速乘
7  {
8      return (x*y-(ll)((long double)x/mod*y)*mod+mod)%mod;
9  }
10 ///////////////////////////////////////////////////////////////////
11 ll mulit(ll a,ll b,ll c){//龟速乘
12     ll ans=0;
13     ll res=a;
14     while(b){
15         if(b&1)
16             ans=(ans+res)%c;
17         res=(res+res)%c;
18         b>>=1;
19     }
20     return ans;
21 }
22 ///////////////////////////////////////////////////////////////////
23 ll pow(ll x,ll n,ll mod){//快速幂
24     ll res=1;
25     while(n>0){
26         if(n%2==1){
27             res=res*x;
28             res=res%mod;
29         }
30         x=x*x;
31         x=x%mod;
32         n>>=1;
33     }
34     return res;
35 }
36 ///////////////////////////////////////////////////////////////////
37 struct matrix{//矩阵快速幂
38     ll m[5][5];
39 };
40 matrix matrix_multi(matrix a,matrix b){
41     matrix tmp;
42     for(int i=0;i<5;i++)
43         for(int j=0;j<5;j++){
44             tmp.m[i][j]=0;
45             for(int k=0;k<5;k++){
46                 tmp.m[i][j]=(tmp.m[i][j])%(MOD-1) + (a.m[i][k]*b.m[k][j]+MOD-1)%(MOD-1))%(MOD-1);
47             }
48         }
49     return tmp;
50 }
51 matrix matrix_pow(matrix a,matrix b,ll n){
52     while(n>0){
53         if(n&1) b=matrix_multi(a,b);
54         a=matrix_multi(a,a);
55         n>>=1;
56     }
57     return b;
58 }
59 ///////////////////////////////////////////////////////////////////
60 ll mulit(ll a,ll b,ll c){//龟速乘
```

IT程序员转型学什么?

关闭

```

60     ll ans=0;
61     ll res=a;
62     while(b){
63         if(b&1)
64             ans=(ans+res)%c;
65         res=(res+res)%c;
66         b>>=1;
67     }
68     return ans;
69 }
70 ll pow(ll x,ll n,ll mod){//快速幂
71     ll res=1;
72     while(n>0){
73         if(n%2==1){
74             res=mulit(res,x,mod);
75         }
76         x=mulit(x,x,mod);
77         n>>=1;
78     }
79     return res;
80 }
81 //////////////////////////////////////
82 ll gcd(ll a,ll b){
83     if(a==0)return 1;
84     if(a<0) return gcd(-a,b);
85     while(b){
86         long long t=a%b;
87         a=b;
88         b=t;
89     }
90     return a;
91 }
92

```

二分优化

```

1  int lower_b(int l,int r,int n){
2      while(l<=r){
3          //cout<<l<<" "<<r<<endl;a
4          int mid=(l+r)/2;
5          if(n>b[mid]) l=mid+1;
6          else r=mid-1;
7      }
8      return l;
9  }
10 int lower_b(int l,int r,int n){
11     while(r-l>1){
12         //cout<<l<<" "<<r<<endl;a
13         int mid=(l+r)/2;
14         if(n>b[mid]) l=mid;
15         else r=mid;
16     }
17     return l;
18 }
19

```

输入挂

```

1  inline int read(){//快读
2  {
3      int x=0,f=1;char c=getchar();
4      while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
5      while(isdigit(c)){x=x*10+c-48;c=getchar();}
6      return x*f;
7  }
8
9  inline void put(int x)//快输
10 {
11     if(x<0) putchar('-'),x=-x;
12     if(x>9) put(x/10);

```

```
13 |      putchar(x%10+'0');  
14 |  }  
15 |
```

文章最后发布于: 2019-10-17 10:51:32

有 0 个人打赏

IT
程序员
转型学
什么？

关闭