

二分图 (6)
二分图匹配 (6)
分块 (3)
分治 (2)
FFT (6)
FNT (1)
FWT (5)
概率DP (2)
概率与期望 (3)
高斯消元 (3)
后缀平衡树 (2)
后缀数组 (1)
后缀自动机 (6)
回文自动机 (3)
基环外向树 (1)
计算几何 (14)
可并堆 (3)
KD树 (2)
KMP (1)
KM算法 (1)
可持久化Trie (1)
可持久化可并堆 (1)
可持久化平衡树 (fhqTreap (1)
括号序列 (1)
Link Cut Tree (6)
拉格朗日乘数法 (1)
乱搞 (14)
Manacher (2)
Miller Rabin & Pollard Rho (3)
模拟退火 (1)
莫比乌斯反演 (13)
莫队 (4)
母函数 (1)
欧拉回路 (1)
偏序集 (1)
其他 (9)
容斥原理 (1)
Simpson积分 (2)
Splay (7)
Stirling数 (0)
StoerWagner (1)
三角剖分 (2)
搜索 (2)
双向队列/单调栈 (1)
数据结构 (29)
数学 (38)
数位DP (3)
树分块 (1)
树链剖分 (9)
树上莫队 (2)
树套树 (4)
斯特林公式 (1)
贪心 (3)
图的绝对中心 (1)
拓扑排序 (1)
网络流 (10)
仙人掌 (3)
线段树 (10)
线段树启发式合并 (1)
线性规划 (2)
线性基 (1)
斜率优化 (1)
虚树 (2)
悬线法 (1)
zkw费用流 (1)

```
26 {
27     int res=1;
28     for (;x>=1, a=a*111*a%Mod)
29         if(x&1) res=res*111*a%Mod;
30     return res;
31 }
32 int Rev[400001];
33
34 void Rader(int n)
35 {
36     for(int i=0;i<(1<<n);i++) Rev[i]=Rev[i>>1]>>1|((i&1)<<(n-1));
37 }
38
39 int FNT(int *a,int n,int d)
40 {
41     Rader(n);
42     int len=1<<n;
43     for(int i=0;i<len;i++) if(Rev[i]>i) swap(a[Rev[i]],a[i]);
44     for(int i=1;i<len;i*=2)
45     {
46         int W=(d==1?Pow(p,D/2/i):Pow(p,D-D/2/i));
47         for(int j=0;j<len;j+=2*i)
48             {
49                 int W0=1;
50                 for(int k=0;k<i;k++)
51                     {
52                         int X=a[j+k],Y=Mul(W0,a[j+k+i]);
53                         a[j+k]=Add(X,Y);a[j+k+i]=Add(X,-Y);
54                         W0=Mul(W0,W);
55                     }
56             }
57     }
58     int In=Pow(len,D-1);
59     if(d==1) for(int i=0;i<len;i++) a[i]=Mul(a[i],In);
60 }
61
62
63 void MUL(int*a,int alen,int *b,int blen,int *M,int &M1)
64 {
65     int n=1;
66     while((1<<n)<=alen+blen)n++;
67     FNT(a,n,1);FNT(b,n,1);
68     for(int i=0;i<(1<<n);i++)
69         M[i]=Mul(a[i],b[i]);
70     FNT(M,n,-1);
71     M1=1<<n;
72     while(M1&&!M[M1-1])M1--;
73 }
74 int Last[100001],Now[100001],L2,Last2[100001];
75 int F[100001],flen,G[100001],glen;
76 int Fact[100001],Inv[100001];
77 void Solve(int n)
78 {
79     if(n==1){Last[0]=0,Last[1]=1;return;}
80     int t=n>>1;
81     Solve(t);
82     flen=t;
83     memset(F,0,sizeof(F));
84     memset(G,0,sizeof(G));
85     memset(Last2,0,sizeof(Last2));
86     for(int i=0;i<=t;i++)
87         F[t-i]=Mul(Last[i],Fact[i]),Last2[i]=Last[i];
88     G[0]=1;
89     for(int i=1;i<=t;i++)
90         G[i]=Mul(Mul(G[i-1],Fact[i-1]),Mul(Inv[i],t));
91     MUL(F,t+1,G,t+1,Now,L2);
92     L2--;
93     for(int i=0;i<=t;i++)
94         Last[t-i]=Mul(Now[i],Inv[t-i]);
95     if(n&1)
96     {
```

关闭

杂文 (14)
折半搜索 (1)
支配树 (1)
置换群 (2)
主席树 (可持久化线段树) (11)
状压DP (3)
字符串处理 (7)
组合数学/群论 (5)
最小生成树 (1)
群论 (1)

文章存档
2017年03月 (5)
2017年02月 (8)
2016年12月 (10)
2016年11月 (10)
2016年10月 (5)
展开

阅读排行
BZOJ1143[CTSC2008]脍 (1808)
BZOJ1146: [CTSC2008] (1749)
懵逼ZJOI2016Round2滚 (1293)
BZOJ3619: [Zjoi2014]瑞 (1242)
BZOJ4092: [Zjoi2015]幻 (1041)
自勉 (892)
BZOJ1208: [HNOI2004] (885)
支配树(Dominator tree) (874)
BZOJ4399: 魔法少女LJJ (867)
BZOJ3262: 陌上花开 CD (847)

评论排行
自勉 (7)
夜底有感 (3)
BZOJ4316: 小C的独立集 (2)
NOI题库_微积分_NOI20 (1)
bzoj4336: BJOI2015 骑士 (1)
AFO (1)
BZOJ3533: [Sdoi2014]向 (1)
堆模版 (0)
后缀数组 (0)
关于凸包 : grahman sca (0)

最新评论
自勉
Chester_King: 后排资辞 , 跪%dalao
自勉
Greninja_Wu: 前排资辞 ! %%%dalao
AFO
qzh_1430586275: 祝文化课顺利
自勉
ruanxingzhi: 加油 !
BZOJ3533: [Sdoi2014]向量集
sjwk2017: QAQ居然上榜了。
自勉
qzh_1430586275: mdzz...傻逼领导死全家
自勉
qq_34985151: 刘天加油——wcy
BZOJ4316: 小C的独立集

```
97         for(int i=t+1;i;i--)
98             Last[i]=Add(Last[i-1],Mul(n-1,Last[i]));
99         Last[0]=Mul(n-1,Last[0]);
100     }
101     MUL(Last,t+2,Last2,n-t+1,Last,L2);
102 }
103
104
105 int main()
106 {
107     int n;
108     scanf("%d",&n);
109     Fact[0]=1;
110     for(int i=1;i<=n;i++)Fact[i]=Mul(Fact[i-1],i);
111     Inv[n]=Pow(Fact[n],Mod-2);
112     for(int i=n-1;~i;i--)
113         Inv[i]=Mul(Inv[i+1],i+1);
114     Solve(n);
115     int l,r;
116     for(int i=0;i<=n;i++)
117         printf("%d ",Last[i]);
118     return 0;
119 }
```

第二类 $Stirling$ 数 :

$S(n,m)$ 表示 n 个元素分成 m 个集合的方案数

那么由定义可得递推式:

$$S(n,m) = S(n,m-1) + S(n-1,m) * m$$

并且

$$k^n = \sum_{m=0}^k A_k^m * S(n,m) = \sum_{m=0}^k m! * C_k^m * S(n,m)$$

可以用 n 个球放入 k 个篮子,允许存在空篮子的方案数

由上式 我们经过二项式反演之后可以得到

$$S(n,m) = \sum_{k=0}^m (-1)^{m-k} * C_m^k * (m-k)^n$$
$$S(n,m) = \sum_{k=0}^m (-1)^k * \frac{1}{k!} * \frac{1}{m-k!} * (m-k)^n$$

同样可以经过FFT ($nlog_2n$)得出

```
1 #include<cstdio>
2 #include<iostream>
3 #include<cstring>
4 #include<cstdlib>
5 #include<cmath>
6 #include<algorithm>
7 using namespace std;
8
9 #define ll long long
10 const
11     int Mod=998244353,p=3,D=Mod-1;
12
13 int Add(int a,int b)
14 {
15     a+=b;
16     if(a>=Mod)a-=Mod;
17     if(a<0)a+=Mod;
18     return a;
19 }
20
21 int Mul(int a,int b)
22 {
23     return a*1ll*b%Mod;
24 }
25
26 int Pow(int a,int x)
27 {
28     int res=1;
29     for(;x;>=1,a=a*1ll*a%Mod)
30         if(x&1)res=res*1ll*a%Mod;
31     return res;
```

Hillan_: @lcrtest:恩?数据没问题吧

BZOJ4316: 小C的独立集

lcrtest: 博主: 图中任何一条边属于且仅属于一个简单环那组数据不合法

一场BC的台前幕后

Hillan_: 太强辣

```

32 }
33
34 int Rev[100001];
35 void Rader(int n)
36 {
37     for(int i=0; i<(1<<n); i++) Rev[i]=(Rev[i>>1]>>1)|((i&1)<<n-1);
38 }
39
40 void FNT(int *a, int n, int d)
41 {
42     Rader(n);
43     int len=1<<n, Inv=Pow(len, Mod-2);
44     for(int i=0; i<len; i++) if(Rev[i]>i) swap(a[Rev[i]], a[i]);
45     for(int i=1; i<len; i*=2)
46     {
47         int W0=(d==1?(Pow(p, D/2/i)): (Pow(p, D-D/2/i)));
48         for(int j=0; j<len; j+=2*i)
49         {
50             int W=1;
51             for(int k=0; k<i; k++)
52             {
53                 int X=a[k+j], Y=Mul(a[k+j+i], W);
54                 a[k+j]=Add(X, Y);
55                 a[k+i+j]=Add(X, -Y);
56                 W=Mul(W, W0);
57             }
58         }
59     }
60     if(d== -1)
61     for(int i=0; i<len; i++) a[i]=Mul(a[i], Inv);
62 }
63
64
65 int F[100001], G[100001], Fact[100001], Inv[100001];
66 int Ans[100001];
67 int main()
68 {
69     int n;
70     scanf("%d", &n);
71     int B=1;
72     while((1<<B)<=n) B++; B++;
73     Fact[0]=1;
74     for(int i=1; i<=n; i++) Fact[i]=Mul(Fact[i-1], i);
75     Inv[n]=Pow(Fact[n], Mod-2);
76     for(int i=n-1; ~i; i--) Inv[i]=Mul(Inv[i+1], i+1);
77     for(int i=0; i<=n; i++)
78     {
79         F[i]=Add(0, Mul((i&1?-1:1), Inv[i]));
80         G[i]=Mul(Pow(i, n), Inv[i]);
81     }
82     FNT(F, B, 1);
83     FNT(G, B, 1);
84     for(int i=0; i<(1<<B); i++)
85     Ans[i]=Mul(F[i], G[i]);
86     FNT(Ans, B, -1);
87     for(int i=0; i<=n; i++)
88     printf("%d ", Ans[i]);
89     return 0;
90 }

```

第二类Stirling数还有一个好性质:

$$x^n = \sum_{i=0}^n A_x^i * S(n, i)$$

依旧考虑x个盒子放n个不同的球,可以存在空盒子

则左式可以看做枚举非空盒子个数 然后按顺序取出i个盒子放入球

根据这个性质我们可以很方便的处理一类代价为n次方的问题

HDU4625 JZPTREE

关闭

给出一棵 n 个节点边权为1的树，对每一个节点 i ，输出

$$\sum_{j=1}^n dist(i, j)^m$$

$$n \leq 5 \times 10^4, m \leq 500$$

注意到此题 m 较小,可以用上述方法求解

唯一的问题就是如何从 A_x^i 得出 A_{x+1}^i

对于 A_{x+1}^i 我们有递推公式

$$A_{x+1}^i = A_x^i + i * A_x^{i-1}$$

然后树形DP处理出每一个点 u 的 A_x^i 和,最后乘上相应的Stirling数即可得到答案

时间复杂度: $O(n * m)$

```

1  #include<cstdio>
2  #include<iostream>
3  #include<cstring>
4  #include<algorithm>
5
6  char c;
7  inline void read(int&a)
8  {
9      a=0;do c=getchar();while(c<'0' || c>'9');
10     while(c<='9' && c>='0') a=(a<<3)+(a<<1)+c-'0', c=getchar();
11 }
12 const
13     int Mod=10007;
14     int S[501][603];
15     int Low[60001][603];
16     int High[60001][603];
17     int n,m;
18     struct Chain
19     {
20         Chain*next;
21         int u;
22     }*Head[50002];
23     Chain R[100001];
24     int A;
25     inline void Add(int a, int b)
26     {Chain*tp=R+A;A++;tp->next=Head[a];Head[a]=tp;tp->u=b;}
27     int F[60001];
28
29     inline int Up(int u, int i)
30     {
31
32         return ((i>=1?i*111*(Low[u][i-1]):0)+(i>=0?Low[u][i]:0))%Mod;
33     }
34
35     inline int add(int a, int b)
36     {
37
38         a+=b;
39         if(a>=Mod) a-=Mod;
40         if(a<0) a+=Mod;
41         return a;
42     }
43
44     inline int mul(int a, int b)
45     {
46
47         a=a*b;
48         a%=Mod;
49         return a;
50     }
51     void DFS(int u, int f)
52     {
53         F[u]=f;
54
55         for (Chain*tp=Head[u]; tp; tp=tp->next)
56             if (tp->u!=f)
57                 {
58

```

```

59     DFS(tp->u, u);
60     for(int i=0; i<=m; i++)
61         Low[u][i+1]=add(Low[u][i+1], Up(tp->u, i+1));
62     Low[u][0]=add(Low[u][0], Low[tp->u][0]);
63
64     }
65     Low[u][0]=add(1, Low[u][0]);
66 }
67
68 void DFS2(int u, int f)
69 {
70     for(Chain*tp=Head[u]; tp; tp=tp->next)
71         if(tp->u!=f)
72         {
73             for(int i=0; i<=m; i++)
74                 High[tp->u][i]
75                 =add(add(add(High[u][i], -Up(tp->u, i)), mul(i, i>=1?add(High[u][i-1], -Up(tp->u, i-1))
76                 DFS2(tp->u, u);
77         }
78
79     }
80
81
82
83 int Calc(int u)
84 {
85     int res=0, f=F[u];
86     if(u==1)
87     {
88         for(int i=1; i<=m; i++)
89         {
90             res=add(res, mul(S[m][i], Low[u][i]));
91         }
92         return res;
93     }
94     for(int i=1; i<=m; i++)
95     {
96         res=
97         add(res,
98             mul(S[m][i],
99                 High[u][i]
100             ));
101     }
102     return res;
103 }
104
105
106
107 int main()
108 {
109     freopen("self.in", "r", stdin);
110     freopen("self.out", "w", stdout);
111     S[0][0]=1;
112     for(int i=1; i<=500; i++)
113         for(int j=1; j<=500; j++)
114             S[i][j]=(S[i-1][j-1]+S[i-1][j]*j)%Mod;
115     int T;
116     read(T);
117     while(T--)
118     {
119         A=0;
120         memset(F, 0, sizeof(F));
121         memset(Low, 0, sizeof(Low));
122         memset(High, 0, sizeof(High));
123         memset(Head, 0, sizeof(Head));
124         read(n), read(m);
125         for(int i=1; i<n; i++)
126         {
127             int a, b;
128             read(a), read(b);
129             Add(a, b), Add(b, a);

```

关闭

```

130     }
131     DFS(1, 1);
132     for(int i=0; i<=m; i++) High[1][i]=Low[1][i];
133     DFS2(1, 1);
134     for(int i=1; i<=n; i++)
135         printf("%d\n", Calc(i));
136
137 }
138 }

```

Hackerrank Costly Graphs

题目大意:

求所有节点为n的图的权值和

这里图G的权值和定义为 $\sum_{u \in G} D(u)^m$

$D(u)$ 为节点u的度

m 为给定常数

$$1 \leq n \leq 10^9$$

$$1 \leq m \leq 2 \cdot 10^5$$

显然每个节点对于答案贡献独立且相等 我们考虑其中任意节点u的贡献

则有贡献

$$W(u) = 2^{C_{n-1}^2} * \sum_{i=0}^{n-1} i^m * C_{n-1}^i$$

$$W(u) = 2^{C_{n-1}^2} * \sum_{i=0}^{n-1} C_{n-1}^i * \sum_{t=0}^m A_i^t * S(m, t)$$

$$W(u) = 2^{C_{n-1}^2} * \sum_{i=0}^{n-1} \frac{(n-1)!}{(n-1-i)!} * \sum_{t=0}^{\min(m, i)} \frac{1}{(i-t)!} * S(m, t)$$

$$W(u) = (n-1)! * 2^{C_{n-1}^2} * \sum_{t=0}^m S(m, t) * \sum_{i=t}^{n-1} \frac{1}{(n-1-i)! * (i-t)!}$$

$$W(u) = (n-1)! * 2^{C_{n-1}^2} * \sum_{t=0}^m S(m, t) * \sum_{i=0}^{n-1-t} \frac{1}{(n-1-t-i)! * (i)!}$$

$$W(u) = (n-1)! * 2^{C_{n-1}^2} * \sum_{t=0}^m S(m, t) * \frac{1}{(n-1-t)!} * 2^{n-1-t}$$

$$W(u) = 2^{C_{n-1}^2} * \sum_{t=0}^m S(m, t) * A_{n-1}^t * 2^{n-1-t}$$

答案就是 $n * W(u)$

注意幕次的取模需要费马小定理..

```

1  #include<cstdio>
2  #include<iostream>
3  #include<cstring>
4  #include<cmath>
5  #include<algorithm>
6  #include<cstdlib>
7  using namespace std;
8
9  const
10     int Mod=1005060097, p=7;
11
12  char c;
13  inline void read(int&a) {a=0;do c=getchar();while(c<'0' || c>'9');while(c<='9' && c>='0') a=(a<
14
15  int Pow(int a, int x)
16  {
17     int res=1;
18     for(;x>=1; a=a*111*a%Mod)
19         if(x&1) res=res*111*a%Mod;
20     return res;
21 }
22 int mul(int a, int b) {return a*111*b%Mod;}
23 int mul(int a, int b, int Mod) {return a*111*b%Mod;}
24 int add(int a, int b)
25 {
26     a=a+b;
27     if(a>=Mod) a-=Mod;
28     if(a<0) a+=Mod;
29     return a;
30 }
31
32 int Rev[1000001];

```

关闭

```

33 inline void Rader(int n)
34 {
35     for(int i=0;i<(1<<n);i++) Rev[i]=(Rev[i>>1]>>1)|((i&1)<<n-1);
36 }
37
38 void FNT(int *a, int n, int l)
39 {
40     int len=1<<n, I=Pow(len, Mod-2);
41     Rader(n);
42     for(int i=0;i<len;i++)
43         if(Rev[i]>i) swap(a[Rev[i]], a[i]);
44     for(int i=1;i<len;i*=2)
45     {
46         int W=Pow(p, l==1?(Mod-1)/2/i:(Mod-1-(Mod-1)/2/i));
47         for(int j=0;j<len;j+=2*i)
48         {
49             int W0=1;
50             for(int k=0;k<i;k++)
51             {
52                 int x=a[j+k], y=mul(a[j+k+i], W0);
53                 a[j+k]=add(x, y);
54                 a[j+k+i]=add(x, -y);
55                 W0=mul(W0, W);
56             }
57         }
58     }
59     if(l==--1)
60         for(int i=0;i<len;i++) a[i]=mul(a[i], I);
61 }
62
63 int Fact[1000001], Inv[1000001];
64 int F[1000001], G[1000001], S[1000001];
65
66 int C2(int n)
67 {
68     if(n&1) return mul(n, n-1>>1, Mod-1);
69     return mul(n>>1, n-1, Mod-1);
70 }
71
72
73 int main()
74 {
75     int T;
76     Fact[0]=1;
77     for(int i=1;i<=200000;i++) Fact[i]=mul(Fact[i-1], i);
78     Inv[200000]=Pow(Fact[200000], Mod-2);
79     for(int i=199999;~i;i--) Inv[i]=mul(Inv[i+1], i+1);
80     read(T);
81     while(T--)
82     {
83         int n, m;
84         read(n), read(m);
85         for(int i=0;i<=m;i++)
86             F[i]=add(0, (i&1?-1:1)*Inv[i]);
87         for(int i=0;i<=m;i++)
88             G[i]=mul(Pow(i, m), Inv[i]);
89         int B=1;
90         while((1<<B)<=m+m+1) B++;
91         FNT(F, B, 1); FNT(G, B, 1);
92         for(int i=0;i<(1<<B);i++)
93             S[i]=mul(F[i], G[i]);
94         FNT(S, B, -1);
95         int ans=0, A=1, To=Pow(2, n-1), I=(Mod>>1)+1;
96         for(int i=0;i<=m;To=mul(To, I), A=mul(A, (n-1-(i++))))
97             ans=add(ans, mul(A, mul(To, S[i])));
98         ans=mul(ans, mul(Pow(2, C2(n-1)), n));
99         printf("%d\n", ans);
100         for(int i=0;i<(1<<B);i++) F[i]=G[i]=S[i]=0;
101     }
102     return 0;
103 }

```

关闭

同时Stirling数可以应用于幂级数..

再来看看小火车在今年NOI十连测上的一道题:

一张无向图的权值为联通块个数的 m 次方, 问所有 n 个点的带标号的无向图的权值和。

答案对998244353取模。

$T \leq 1000, n \leq 30000, m \leq 15$ 。

构造一个多项式 $L = \sum_{i=1}^n x_i$

其中 x_i 表示连通块 i 是否存在

则 L^m 为该图的权值

由扩展二项式定理可得

项 $\prod x_{a_i}^{t_i}$ 的系数为 $\frac{m!}{\prod t_i!}$, 其中 $\sum t_i = m$

考虑所有由 x_{a_1} 至 x_{a_k} 组成的项的系数和

对于任意 $\sum_{i=1}^k t_i = m$

我们有 $\sum \frac{m!}{\prod t_i!} = \sum S(m, k) * k!$

则对于某 k 个联通块, 如果它们在图中同时出现, 那么贡献为

$S(m, k) \times k!$

设 f_i 为 i 个点的连通图数目

$f_i = 2^{C_i^2} - \sum_{j=1}^{i-1} C_{i-1}^{j-1} * f_j * 2^{C_{i-j}^2}$

我们可以使用多项式求逆的方法求出 f

设 $g_{i,j}$ 为 i 个点 j 个块的无向图个数

有 $g_{i,j} = \sum_{k=1}^i C_i^k * f_k * g_{i-k,j-1}$

可以用卷积算出

答案即为 $\sum_{j=1}^m j! * S(m, j) \sum_{i=1}^n C_n^i * g_{i,j} * 2^{C_{n-i}^2}$

可以把 $\sum_{i=1}^n C_n^i * g_{i,j} * 2^{C_{n-i}^2}$ 看成 $T(j, n)$

m 次卷积预处理之后 $O(m)$ 累加即可

时间复杂度 $O(Tm + mn \log_2 n)$

多项式求逆的话具体看2015年集训队鏖爷论文

自己的代码常数巨大

```

1  #include<cstdio>
2  #include<iostream>
3  #include<cmath>
4  #include<cstring>
5  using namespace std;
6  const
7      int Mod=998244353, p=3, D=Mod-1;
8
9
10 int Pow(int a, int x)
11 {
12     int res=1;
13     for (; x>=1, a=a*111*a%Mod)
14         if (x&1) res=res*111*a%Mod;
15     return res;
16 }
17 inline
18 int Mul(int a, int b)
19 {return a*111*b%Mod;}
20 inline int Add(int a, int b)
21 {
22     a+=b;
23     if (a>=Mod) a-=Mod;
24     if (a<0) a+=Mod;
25     return a;
26 }
27
28 int Rev[2000001];
29 int Rader(int n)

```

关闭

```

30 {
31     for(int i=0;i<1<<n;i++) Rev[i]=(Rev[i>>1]>>1)|((i&1)<<n-1);
32 }
33
34 int FNT(int *a,int n,int f)
35 {
36     int len=1<<n,I=Pow(len,D-1);
37     Rader(n);
38     for(int i=0;i<len;i++) if(Rev[i]>i) swap(a[Rev[i]],a[i]);
39     for(int i=1;i<len;i*=2)
40     {
41         int W=Pow(p,f==1?D/2/i:(D-D/2/i));
42         for(int j=0;j<len;j+=i*2)
43         {
44             int W0=1;
45             for(int k=0;k<i;k++)
46             {
47                 int x=a[j+k],y=Mul(W0,a[i+j+k]);
48                 a[j+k]=Add(x,y),a[i+j+k]=Add(x,-y);
49                 W0=Mul(W0,W);
50             }
51         }
52     }
53     if(f==1)
54     for(int i=0;i<len;i++) a[i]=Mul(a[i],I);
55 }
56
57 int T[200001],A[200001];
58 int Fact[100001],Inv[100001];
59
60 int C(int x)
61 {
62     return (x*1ll*(x-1)>>1)%D;
63 }
64
65 int B[200001],Ca[200001];
66
67 void Find(int Len)
68 {
69     if(Len==1)
70         {B[0]=1;return;}
71     int t=1;
72     Find(Len>>1);
73     memset(Ca,0,sizeof(Ca));
74     memset(A,0,sizeof(A));
75     while((1<<t)<=3*Len) t++;
76     for(int i=0;i<Len;i++)
77         A[i]=T[i];
78     FNT(A,t,1);
79     FNT(B,t,1);
80     for(int i=0;i<1<<t;i++)
81         Ca[i]=Add(Mul(2,B[i]),-Mul(B[i],Mul(B[i],A[i])));
82     FNT(A,t,-1);
83     FNT(Ca,t,-1);
84     FNT(B,t,-1);
85     memset(B,0,sizeof(B));
86     for(int i=0;i<Len;i++)
87         B[i]=Ca[i];
88 }
89
90 int S[101][101];
91
92 int G[2][100001];
93
94 int F[100001];
95 int ANS[101][100001];
96 int BIT[100001];
97 int main()
98 {
99     freopen("self.in","r",stdin);
100     freopen("self.out","w",stdout);

```

关闭

```

101
102     int N=16384*2-1;
103     Fact[0]=1;
104     int P=116195171;
105     for(int i=1;i<=N;i++)Fact[i]=Mul(i,Fact[i-1]);
106     Inv[N]=Pow(Fact[N],D-1);
107     for(int i=N-1;~i;i--)Inv[i]=Mul(i+1,Inv[i+1]);
108     for(int i=1;i<=N;i++)
109         T[i]=Mul(Pow(2,C(i)),Inv[i]);
110     T[0]=1;
111     Find(N+1);
112     int TTT=16;
113     FNT(B,TTT,1);
114     memset(T,0,sizeof(T));
115     for(int i=1;i<=N;i++)
116         T[i]=Mul(Inv[i-1],Pow(2,C(i)));
117     FNT(T,TTT,1);
118     for(int i=0;i<1<<TTT;i++)
119         F[i]=Mul(T[i],B[i]);
120     FNT(F,TTT,-1);
121     B[0]=0;
122     int cas;
123     S[1][1]=1;
124     for(int i=2;i<=100;i++)
125     for(int j=1;j<=100;j++)
126         S[i][j]=Add(S[i-1][j-1],Mul(j,S[i-1][j]));
127
128
129     N=1;
130     int n=16384*2,m=15,now=1,next=0;
131     G[now][1]=1;
132     while((1<<N)<n)N++;
133     N++;
134
135     for(int i=1;i<=n;i++)
136         G[now][i]=Mul(F[i],Fact[i-1]);
137     memset(T,0,sizeof(T));
138     for(int i=1;i<=n;i++)
139         T[i]=F[i];
140     T[0]=0;
141     FNT(T,N,1);
142     for(int i=1;i<=n;i++)
143         BIT[i]=Mul(Inv[i],Pow(2,C(i)));
144     BIT[0]=1;
145     FNT(BIT,N,1);
146     for(int j=1;j<=m;j++,next^=1,now^=1)
147     {
148         for(int i=1;i<=n;i++)
149             G[now][i]=Mul(G[now][i],Inv[i]);
150         FNT(G[now],N,1);
151         for(int i=0;i<1<<N;i++)
152             ANS[j][i]=Mul(BIT[i],G[now][i]);
153         FNT(ANS[j],N,-1);
154         memset(G[next],0,sizeof(G[next]));
155         for(int i=0;i<1<<N;i++)
156             G[next][i]=Mul(G[now][i],T[i]);
157         FNT(G[next],N,-1);
158         for(int i=n+1;i<1<<N;i++)
159             G[next][i]=0;
160         for(int i=1;i<=n;i++)
161             G[next][i]=Mul(G[next][i],Fact[i-1]);
162         G[next][0]=0;
163     }
164
165
166     scanf("%d",&cas);
167     while(cas--)
168     {
169         int ans=0;
170         int n,m;
171         memset(G,0,sizeof(G));scanf("%d%d",&n,&m);

```

关闭

```

172         for(int j=1; j<=m; j++)
173             ans=Add(ans, Mul(S[m][j], Mul(Fact[n], Mul(Fact[j], ANS[j][n]))));
174         printf("%d\n", ans);
175     }
176     return 0;
177 }

```

还有利用反演的:

$$f_i = \sum_{j=0}^i S(i, j) g_j$$

为

$$g_i = \sum_{j=0}^i (-1)^{i-j} s(i, j) f_j$$

的充要条件

然后容斥的这里就埋个坑...什么时候会做了再说吧

关于容斥:

满足任意两行或者任意两列都不相同的 $n \times m$ 的数字矩阵有多少个 ,
 每一个格子内的数必须是 $[1, C]$ 内的整数。对一个大质数 $(10^9 + 7)$ 取模。

$$n, m, C \leq 4000$$

首先考虑如何容斥..

f_i 表示列上最多有 i 个等价类每行不重复的方案数

g_i 表示列上刚好有 i 个等价类每行不重复的方案数

$$\text{则有 } f_i = S(m, i) * A_{m,i}^n = \sum_{j=1}^i \frac{g_j}{S(m, j)} * S(m, i)$$

然后就可以 $O(m^2)$ 暴力啦

```

1  #include<cstdio>
2  #include<cstring>
3  #include<cstdlib>
4  #include<cmath>
5
6  using namespace std;
7  const
8      int Mod=1000000007;
9  #define ll long long
10 inline int Mul(int a, int b)
11 {
12     return a*ll*b%Mod;
13 }
14
15 inline int Add(int a, int b)
16 {
17     a+=b;
18     if(a>=Mod) a-=Mod;
19     if(a<0) a+=Mod;
20     return a;
21 }
22
23 inline int Pow(int a, int x)
24 {
25     int res=1;
26     for(; x>=1; a=a*ll*a%Mod)
27         if(x&1) res=res*ll*a%Mod;
28     return res;
29 }
30
31 int F[10001], G[10001];
32 int S[4001][4001];
33 class CountTables{
34 public:
35     int howMany(int n, int m, int C)
36     {
37         S[1][1]=1;
38         for(int i=2; i<=4000; i++)
39             for(int j=1; j<=4000; j++) S[i][j]=Add(S[i-1][j-1], Mul(j, S[i-1][j]));
40         int INV=1;
41         INV=Pow(INV, Mod-2);

```

关闭

```
42     int D=C, P=1;
43     for (int i=1; i<=m; i++)
44     {
45         int V=1;
46         for (int j=0; j<n; j++)
47             V=Mul (V, Add (D, -j)) ;
48         F[i]=V;
49         D=Mul (D, C) ;
50         P=Mul (P, i) ;
51     }
52     for (int i=1; i<=m; i++)
53     {
54         G[i]=F[i];
55         for (int j=1; j<i; j++)
56             G[i]=Add (G[i], -Mul (G[j], S[i][j]));
57     }
58     return Mul (G[m], 1) ;
59     return Mul (G[m], Pow (INV, Mod-2)) ;
60 } } Gh;
61
62
63 int main ()
64 {
65     int n, m, C;
66     scanf ("%d%d%d", &n, &m, &C) ;
67     printf ("%d\n", Gh. howMany (n, m, C)) ;
68 }
```

To Be continued

顶 0 踩 0

上一篇 Hackerrank XOR Subsequences
下一篇 HDU5766 Filling

相关文章推荐

- [组合数学] 第一类，第二类Stirling数，Bell数
- 【免费】搜狗机器翻译技术分享
- 第一类Stirling数和第二类Stirling
- 深度学习在推荐领域的应用和实践--吴岸城
- 斯特林数
- Git和Github快速上手指南
- [组合数学] 第一类，第二类Stirling数，Bell数
- JFinal专题之模板引擎知识集锦
- boj 1336 简单的问题 不过自己没想到 别人解释的...
- Spring Cloud 微服务实战
- AVR学习笔记三、定时计数器0实验
- JDK9新特性解答
- Stirling 数
- 【Fibonacci 序列+第一类Stirling数+二项式定理】 ...
- 山东省第五届ACM大学生程序设计竞赛 Hearthsto...
- [组合数学]

关闭

查看评论

暂无评论

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

