

Algorithms Crush Me

博客园

首页

新随笔

联系

订阅


管理


随笔 - 124 文章 - 0 评论 - 26

The 2018 ACM-ICPC Chinese Collegiate Programming Contest


这个拆队比赛的题目质量优秀过头了吧？

A.Maximum Element In A Stack







```
1 #include <bits/stdc++.h>
2 using namespace std;
3 const int maxn = 5e6 + 10;
4 typedef long long LL;
5
6 int n, p, q, m;
7 unsigned int SA, SB, SC;
8 unsigned int rng61() {
9     SA ^= SA << 16;
10    SA ^= SA >> 5;
11    SA ^= SA << 1;
12    unsigned int t = SA;
13    SA = SB;
14    SB = SC;
15    SC ^= t ^ SA;
16    return SC;
17 }
18 stack<int> st, ma;
19 LL gen() {
20     LL ret = 0;
21     while(!st.empty()) st.pop(), ma.pop();
22     st.push(0), ma.push(0);
23     scanf("%d%d%d%d%u%u", &n, &p, &q, &m, &SA, &SB, &SC);
24     for (int i = 1; i <= n; i++) {
25         if (rng61() % (p + q) < p) st.push(rng61() % m + 1), ma.push(max(st.top(),
ma.top()));
26         else if(st.size() > 1) st.pop(), ma.pop();
27         ret ^= (LL) i * ma.top();
28     }
29     return ret;
30 }
31 int main() {
32     int T;
33     scanf("%d", &T);
34     for(int kase = 1; kase <= T; ++kase) {
35         printf("Case #d: %lld\n", kase, gen());
36     }
37     return 0;
38 }
```



B.Rolling The Polygon





```
1 #include <bits/stdc++.h>
2 using namespace std;
```

公告

昵称: Aguin
园龄: 4年3个月
粉丝: 35
关注: 21
[+加关注](#)

搜索

随笔档案

- 2019年3月(2)
- 2018年8月(4)
- 2018年7月(6)
- 2018年6月(7)
- 2018年5月(7)
- 2018年4月(10)
- 2017年10月(1)
- 2017年7月(2)
- 2017年3月(1)
- 2017年2月(1)
- 2017年1月(2)
- 2016年12月(1)
- 2016年10月(2)
- 2016年9月(4)
- 2016年8月(2)
- 2016年7月(8)
- 2016年5月(1)
- 2016年4月(2)
- 2016年3月(3)
- 2016年2月(4)
- 2016年1月(5)
- 2015年12月(5)
- 2015年11月(4)
- 2015年10月(4)
- 2015年9月(9)
- 2015年8月(11)
- 2015年7月(8)
- 2015年6月(4)
- 2015年5月(4)

```

3 double x[111], y[111];
4
5 const double pi = acos(-1);
6 double sqr(double x) {return x * x;}
7 double dis(int i, int j) {
8     return sqrt(sqr(x[i] - x[j]) +sqr(y[i] - y[j]));
9 }
10
11 int main() {
12     int T;
13     scanf("%d", &T);
14     for(int kase = 1; kase <= T; ++kase) {
15         int n;
16         scanf("%d", &n);
17         for(int i = 0; i <= n; ++i) scanf("%lf %lf", x + i, y + i);
18         double ans = 0;
19         for(int j = 0; j < n; ++j) {
20             int i = (j - 1 + n) % n, k = (j + 1) % n;
21             double d1 = dis(i, j), d2 = dis(j, k), d3 = dis(i, k);
22             double cos_theta = (sqr(d1) + sqr(d2) -sqr(d3)) / 2 / d1 / d2, theta =
acos(cos_theta);
23             ans += dis(j, n) * (pi - theta);
24         }
25         printf("Case #d: %.3f\n", kase, ans);
26     }
27     return 0;
28 }

```

C.Caesar Cipher

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 char A[55], B[55], C[55];
4
5 int main() {
6     int T;
7     scanf("%d", &T);
8     for(int kase = 1; kase <= T; ++kase) {
9         int n, m;
10        scanf("%d %d %s %s %s", &n, &m, A + 1, B + 1, C + 1);
11        int b = (B[1] - A[1] + 26) % 26;
12        for(int i = 1; i <= m; ++i) C[i] = (C[i] - 'A' - b + 26) % 26 + 'A';
13        printf("Case #d: ", kase);
14        puts(C + 1);
15    }
16    return 0;
17 }

```

D.Take Your Seat

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int T;
6     scanf("%d", &T);
7     for(int kase = 1; kase <= T; ++kase) {
8         int n, m;
9         scanf("%d %d", &n, &m);
10        printf("Case #d: %.6f %.6f\n", kase, n == 1 ? 1 : 0.5, (m + 1) / 2.0 / m);
11    }

```

```

12     return 0;
13 }

```

E.2-3-4 Tree

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int maxn = 1e5 + 10;
4 int a[maxn];
5
6 int cnt, rt;
7 struct node {
8     int f;
9     vector<int> d, ch;
10    void init(int fa, int x) {
11        f = fa;
12        d.clear(), ch.clear();
13        d.push_back(x);
14    }
15 } tr[maxn];
16
17 void ins(int p, int x) {
18     if(tr[p].d.size() == 3) {
19         int f = tr[p].f;
20         vector<int> cpy_d, cpy_ch;
21         cpy_d.swap(tr[p].d), cpy_ch.swap(tr[p].ch);
22         if(p == rt) {
23             tr[rt==++cnt].init(0, cpy_d[1]);
24             tr[++cnt].init(rt, cpy_d[0]);
25             tr[p].init(rt, cpy_d[2]);
26             tr[rt].ch.push_back(cnt), tr[rt].ch.push_back(p);
27             if(cpy_ch.size()) {
28                 tr[cnt].ch.push_back(cpy_ch[0]), tr[cpy_ch[0]].f = cnt;
29                 tr[cnt].ch.push_back(cpy_ch[1]), tr[cpy_ch[1]].f = cnt;
30                 tr[p].ch.push_back(cpy_ch[2]), tr[cpy_ch[2]].f = p;
31                 tr[p].ch.push_back(cpy_ch[3]), tr[cpy_ch[3]].f = p;
32             }
33             p = rt;
34         }
35         else {
36             tr[p].init(f, cpy_d[0]);
37             tr[++cnt].init(f, cpy_d[2]);
38             tr[f].d.push_back(cpy_d[1]);
39             sort(tr[f].d.begin(), tr[f].d.end());
40             tr[f].ch.push_back(cnt);
41             for(int i = tr[f].ch.size() - 1; i > 1; --i) {
42                 if(tr[f].ch[i-1] != p) swap(tr[f].ch[i-1], tr[f].ch[i]);
43                 else break;
44             }
45             if(cpy_ch.size()) {
46                 tr[p].ch.push_back(cpy_ch[0]), tr[cpy_ch[0]].f = p;
47                 tr[p].ch.push_back(cpy_ch[1]), tr[cpy_ch[1]].f = p;
48                 tr[cnt].ch.push_back(cpy_ch[2]), tr[cpy_ch[2]].f = cnt;
49                 tr[cnt].ch.push_back(cpy_ch[3]), tr[cpy_ch[3]].f = cnt;
50             }
51             p = f;
52         }
53     }
54     if(tr[p].ch.size() == 0) {
55         tr[p].d.push_back(x);
56         sort(tr[p].d.begin(), tr[p].d.end());
57     }
58     else {
59         if(x < tr[p].d[0]) ins(tr[p].ch[0], x);
60         else if(x > tr[p].d[tr[p].d.size()-1]) ins(tr[p].ch[tr[p].ch.size()-1], x);
61         else {
62             for(int i = 1; i < tr[p].d.size(); ++i)
63                 if(x < tr[p].d[i]) {ins(tr[p].ch[i], x); break;}

```

```

64     }
65 }
66 }
67
68 void dfs(int p) {
69     for(int i = 0; i < tr[p].d.size(); ++i)
70         printf("%d%c", tr[p].d[i], i == tr[p].d.size() - 1 ? '\n' : ' ');
71     for(int i = 0; i < tr[p].ch.size(); ++i)
72         dfs(tr[p].ch[i]);
73 }
74
75 int main() {
76     int T;
77     scanf("%d", &T);
78     for(int kase = 1; kase <= T; ++kase) {
79         int n;
80         scanf("%d", &n);
81         for(int i = 1; i <= n; ++i) scanf("%d", &a[i]);
82         cnt = rt = 1, tr[rt].init(0, a[1]);
83         for(int i = 2; i <= n; ++i) ins(rt, a[i]);
84         printf("Case %d:\n", kase);
85         dfs(rt);
86     }
87     return 0;
88 }

```

F.Moving On

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int G[222][222], w[222], nid[222];
4 int u[22222], v[22222], qw[22222], qid[22222], ans[22222];
5 bool cmp1(int i, int j) {return w[i] < w[j];}
6 bool cmp2(int i, int j) {return qw[i] < qw[j];}
7 void floyd(int x, int n) {
8     for(int i = 1; i <= n; ++i)
9         for(int j = 1; j <= n; ++j)
10             G[i][j] = min(G[i][j], G[i][x] + G[x][j]);
11 }
12 int main() {
13     int T;
14     scanf("%d", &T);
15     for(int kase = 1; kase <= T; ++kase) {
16         int n, q;
17         scanf("%d %d", &n, &q);
18         for(int i = 1; i <= n; ++i) scanf("%d", &w[i]), nid[i] = i;
19         sort(nid + 1, nid + 1 + n, cmp1);
20         for(int i = 1; i <= n; ++i)
21             for(int j = 1; j <= n; ++j)
22                 scanf("%d", &G[i][j]);
23         for(int i = 1; i <= q; ++i) scanf("%d %d %d", &u[i], &v[i], &qw[i]), qid[i] = i;
24         sort(qid + 1, qid + 1 + q, cmp2);
25         int p = 0;
26         for(int i = 1; i <= q; ++i) {
27             while(p < n && w[nid[p+1]] <= qw[qid[i]]) floyd(nid[++p], n);
28             ans[qid[i]] = G[u[qid[i]]][v[qid[i]]];
29         }
30         printf("Case %d:\n", kase);
31         for(int i = 1; i <= q; ++i) printf("%d\n", ans[i]);
32     }
33     return 0;
34 }

```

G.Factories

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int maxn = 1e5 + 10;
4 vector<int> G[maxn], D[maxn];
5 int n, k;
6
7 typedef long long LL;
8 const LL INF = 1e18;
9 LL f[maxn][105];
10 int leaf[maxn];
11 void dfs(int x, int fa, int w) {
12     leaf[x] = 0;
13     for(int i = 1; i <= k; ++i) f[x][i] = INF;
14     if(G[x].size() == 1) f[x][1] = (LL) w * (k - 1), leaf[x]++;
15     for(int i = 0; i < G[x].size(); ++i) {
16         int to = G[x][i], nw = D[x][i];
17         if(to == fa) continue;
18         dfs(to, x, nw);
19         for(int j = min(k, leaf[x] + leaf[to]); j >= 0; --j)
20             for(int p = max(0, j - leaf[x]); p <= min(leaf[to], j); ++p)
21                 f[x][j] = min(f[x][j], f[x][j-p] + f[to][p] + (LL) w * (j * (k - j) - (j - p) * (k - j + p)));
22         leaf[x] += leaf[to];
23     }
24 }
25
26 int main() {
27     int T;
28     scanf("%d", &T);
29     for(int kase = 1; kase <= T; ++kase) {
30         scanf("%d %d", &n, &k);
31         for(int i = 1; i <= n; ++i) G[i].clear(), D[i].clear();
32         for(int i = 1; i < n; ++i) {
33             int u, v, w;
34             scanf("%d %d %d", &u, &v, &w);
35             G[u].push_back(v), D[u].push_back(w);
36             G[v].push_back(u), D[v].push_back(w);
37         }
38         if(n == 2) {
39             printf("Case #%d: %d\n", kase, k == 2 ? D[1][0] : 0);
40             continue;
41         }
42         int rt = 0;
43         for(int i = 1; i <= n; ++i)
44             if(G[i].size() >= 2) rt = i;
45         dfs(rt, 0, 0);
46         printf("Case #%d: %lld\n", kase, f[rt][k]);
47     }
48     return 0;
49 }

```

H.Fight Against Monsters

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long LL;
4 const int maxn = 1e5 + 10;
5 int a[maxn], b[maxn], c[maxn], d[maxn];
6 bool cmp(int i, int j) {return (LL) c[j] * b[i] > c[i] * b[j];}
7 int main() {
8     int T;
9     scanf("%d", &T);
10    for(int kase = 1; kase <= T; ++kase) {
11        int n;
12        scanf("%d", &n);

```

```

13     for(int i = 1; i <= n; ++i) {
14         scanf("%d %d", a + i, b + i);
15         int sum = 0;
16         for(int j = 1; ; ++j) {
17             sum += j;
18             if(sum >= a[i]) {c[i] = j; break;}
19         }
20         d[i] = i;
21     }
22     sort(d + 1, d + 1 + n, cmp);
23     LL ans = 0, sum = 0;
24     for(int i = 1; i <= n; ++i) {
25         int x = d[i];
26         sum += c[x];
27         ans += sum * b[x];
28     }
29     printf("Case #d: %lld\n", kase, ans);
30 }
31 return 0;
32 }

```

I. Bubble Sort

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long LL;
4 LL po[55];
5
6 int main() {
7     int T;
8     scanf("%d", &T);
9     for(int kase = 1; kase <= T; ++kase) {
10         int n, k, q;
11         scanf("%d %d %d", &n, &k, &q);
12         if(k >= n - 1) {
13             LL ans = 1;
14             for(int i = 1; i <= n; ++i) ans = ans * i % q;
15             printf("Case #d: %lld\n", kase, ans);
16             continue;
17         }
18         po[0] = 1;
19         for(int i = 1; i <= n; ++i) po[i] = po[i-1] * (k + 1) % q;
20         LL ans = (po[n-k] + (LL) (n - k - 1) * (n - k) / 2 % q * po[n-k-1]) % q;
21         for(int i = 1; i <= n - k - 2; ++i) ans = (ans + po[i] * i) % q;
22         for(int i = 1; i <= k; ++i) ans = ans * i % q;
23         printf("Case #d: %lld\n", kase, ans);
24     }
25     return 0;
26 }

```

J. Nested Triangles

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long LL;
4 const int maxn = 1e5 + 10;
5 LL x[maxn], y[maxn], xp, yp, xq, yq;
6 LL cross(LL x1, LL y1, LL x2, LL y2) {
7     return x1 * y2 - x2 * y1;
8 }
9 bool cmp1(int i, int j) {
10     return cross(x[i] - xp, y[i] - yp, x[j] - xp, y[j] - yp) > 0;

```

```

11 }
12 bool cmp2(int i, int j) {
13     return cross(x[i] - xq, y[i] - yq, x[j] - xq, y[j] - yq) < 0;
14 }
15
16 int a[maxn], b[maxn], c[maxn], f[maxn], pre[maxn];
17 int lowbit(int s) {
18     return s & (-s);
19 }
20 void modify(int i, int x) {
21     while (i < maxn) {
22         if(x != -1 && (c[i] == -1 || f[x] >= f[c[i]])) {
23             if(c[i] == -1 || f[x] > f[c[i]]) c[i] = x;
24             else if(x < c[i]) c[i] = x;
25         }
26         i += lowbit(i);
27     }
28     return;
29 }
30 int query(int i) {
31     int ret = -1;
32     while (i > 0) {
33         if(c[i] != -1 && (ret == -1 || f[c[i]] >= f[ret])) {
34             if(ret == -1 || f[c[i]] > f[ret]) ret = c[i];
35             else if(c[i] < ret) ret = c[i];
36         }
37         i -= lowbit(i);
38     }
39     return ret;
40 }
41 const int INF = 1e9;
42 void print(int x) {
43     while(x != -1) {
44         printf("%d\n", x);
45         x = pre[x];
46     }
47 }
48 vector<int> id[2];
49 int main() {
50     int T;
51     scanf("%d", &T);
52     for(int kase = 1; kase <= T; ++kase) {
53         scanf("%lld %lld %lld %lld", &xp, &yp, &xq, &yq);
54         int n;
55         scanf("%d", &n);
56         id[0].clear(), id[1].clear();
57         for(int i = 1; i <= n; ++i) {
58             scanf("%lld %lld", x + i, y + i);
59             if(cross(xq - xp, yq - yp, x[i] - xp, y[i] - yp) > 0) id[0].push_back(i);
60             else id[1].push_back(i);
61         }
62         int M[2];
63         for(int i = 0; i <= 1; ++i) {
64             sort(id[i].begin(), id[i].end(), cmp1);
65             for(int j = 0; j < id[i].size(); ++j) {
66                 if(j == 0) a[id[i][j]] = 1;
67                 else if(cmp1(id[i][j-1], id[i][j])) a[id[i][j]] = a[id[i][j-1]] + 1;
68                 else a[id[i][j]] = a[id[i][j-1]];
69             }
70             sort(id[i].begin(), id[i].end(), cmp2);
71             for(int j = 0; j < id[i].size(); ++j) {
72                 if(j == 0) b[id[i][j]] = 1;
73                 else if(cmp2(id[i][j-1], id[i][j])) b[id[i][j]] = b[id[i][j-1]] + 1;
74                 else b[id[i][j]] = b[id[i][j-1]];
75             }
76             M[i] = 0;
77             for(int j = 0; j <= id[i].size(); ++j) c[j] = -1;
78             for(int j = 0; j < id[i].size(); ++j) {
79                 int k = j;
80                 while(k + 1 < id[i].size() && b[id[i][k+1]] == b[id[i][k]]) k++;
81                 for(int p = j; p <= k; ++p) {
82                     int o = id[i][p];
83                     pre[o] = query(a[o] - 1);

```

```

84         f[o] = pre[o] == -1 ? 1 : f[pre[o]] + 1;
85         M[i] = max(M[i], f[o]);
86     }
87     for(int p = j; p <= k; ++p) {
88         int o = id[i][p];
89         modify(a[o], o);
90     }
91     j = k;
92 }
93 swap(xp, xq), swap(yp, yq);
94 }
95 printf("Case #%d: %d\n", kase, max(M[0], M[1]));
96 for(int i = 1; i <= n; ++i) {
97     if(f[i] == max(M[0], M[1])) {
98         print(i); break;
99     }
100 }
101 }
102 return 0;
103 }

```



K.Vertex Covers

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long LL;
4 LL a[55], f[1<<20];
5 vector<int> G[55];
6
7 int main() {
8     int T;
9     scanf("%d", &T);
10    for(int kase = 1; kase <= T; ++kase) {
11        int n, m, q;
12        scanf("%d %d %d", &n, &m, &q);
13        for(int i = 1; i <= n; ++i) scanf("%lld", a + i), G[i].clear();
14        for(int i = 1; i <= m; ++i) {
15            int u, v;
16            scanf("%d %d", &u, &v);
17            G[u].push_back(v);
18            G[v].push_back(u);
19        }
20        if(n == 1) {printf("Case #%d: %lld\n", kase, (1 + a[1]) % q); continue;}
21        int k1 = n / 2, k2 = n - k1;
22        for(int i = 0; i < (1 << k1); ++i) {
23            int ok = 1;
24            LL sum = 1;
25            for(int j = 1; j <= k1; ++j) {
26                if(i & (1 << (j - 1))) sum = sum * a[j] % q;
27                for(int k = 0; k < G[j].size(); ++k) {
28                    int to = G[j][k];
29                    if(to > k1) continue;
30                    if(!(i & (1 << (j - 1))) && !(i & (1 << (to - 1)))) {ok = 0; break;}
31                }
32                if(!ok) break;
33            }
34            f[i] = ok ? sum : 0;
35        }
36        for(int i = 0; i < k1; ++i)
37            for(int j = 0; j < (1 << k1); ++j)
38                if(j & (1 << i)) f[j^(1<<i)] = (f[j^(1<<i)] + f[j]) % q;
39        LL ans = 0;
40        for(int i = 0; i < (1 << k2); ++i) {
41            int ok = 1, msk = 0;
42            LL sum = 1;
43            for(int j = k1 + 1; j <= n; ++j) {
44                if(i & (1 << (j - k1 - 1))) sum = sum * a[j] % q;
45                for(int k = 0; k < G[j].size(); ++k) {

```



```

46         int to = G[j][k];
47         if(to > k1) {
48             if(!(i & (1 << (j - k1 - 1))) && !(i & (1 << (to - k1 - 1)))) {ok
= 0; break;}
49         }
50         else if(!(i & (1 << (j - k1 - 1)))) msk |= 1 << (to - 1);
51     }
52     if(!ok) break;
53 }
54     if(ok) ans = (ans + sum * f[msk]) % q;
55 }
56     printf("Case #d: %lld\n", kase, ans);
57 }
58     return 0;
59 }

```

L.Continuous Intervals

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int maxn = 1e5 + 10;
4 typedef long long LL;
5
6 // seg
7 int M[maxn<<2], cnt[maxn<<2], tag[maxn<<2];
8 void gather(int p) {
9     M[p] = min(M[p << 1], M[p << 1 | 1]);
10    if (M[p << 1] == M[p << 1 | 1]) cnt[p] = cnt[p << 1] + cnt[p << 1 | 1];
11    else if (M[p << 1] < M[p << 1 | 1]) cnt[p] = cnt[p << 1];
12    else cnt[p] = cnt[p << 1 | 1];
13 }
14 void push(int p) {
15     if (tag[p]) {
16         tag[p << 1] += tag[p];
17         tag[p << 1 | 1] += tag[p];
18         M[p << 1] += tag[p];
19         M[p << 1 | 1] += tag[p];
20         tag[p] = 0;
21     }
22 }
23 void build(int p, int l, int r)
24 {
25     tag[p] = 0, cnt[p] = 1;
26     if(l < r)
27     {
28         int mid = (l + r) >> 1;
29         build(p<<1, l, mid);
30         build(p<<1|1, mid + 1, r);
31         gather(p);
32     }
33     else M[p] = 0;
34 }
35 void modify(int p, int tl, int tr, int l, int r, int v) {
36     if (tl > tr) return;
37     if (tr < 1 || r < tl) return;
38     if (l <= tl && tr <= r) {
39         tag[p] += v;
40         M[p] += v;
41         return;
42     }
43     push(p);
44     int mid = (tl + tr) >> 1;
45     modify(p << 1, tl, mid, l, r, v);
46     modify(p << 1 | 1, mid + 1, tr, l, r, v);
47     gather(p);
48 }
49 int query(int p, int tl, int tr, int l, int r) {
50     if (tl > tr) return 0;

```

```

51     if (tr < 1 || r < tl) return 0;
52     if (l <= tl && tr <= r) return M[p] == 0 ? cnt[p] : 0;
53     push(p);
54     int mid = (tl + tr) >> 1;
55     return query(p << 1, tl, mid, l, r) + query(p << 1 | 1, mid + 1, tr, l, r);
56 }
57
58 int a[maxn];
59 map<int, int> pre;
60 stack<int> up, down;
61 int main() {
62     int T;
63     scanf("%d", &T);
64     for(int kase = 1; kase <= T; ++kase) {
65         int n;
66         scanf("%d", &n);
67         build(1, 1, n);
68         pre.clear();
69         while(!up.empty()) up.pop();
70         while(!down.empty()) down.pop();
71         LL ans = 0;
72         for(int i = 1; i <= n; ++i) {
73             scanf("%d", &a[i]);
74             while(up.size() && a[up.top()] > a[i]) {
75                 int t = up.top(); up.pop();
76                 if(up.size()) modify(1, 1, n, up.top() + 1, t, a[t] - a[i]);
77                 else modify(1, 1, n, 1, t, a[t] - a[i]);
78             }
79             while(down.size() && a[down.top()] < a[i]) {
80                 int t = down.top(); down.pop();
81                 if(down.size()) modify(1, 1, n, down.top() + 1, t, a[i] - a[t]);
82                 else modify(1, 1, n, 1, t, a[i] - a[t]);
83             }
84             up.push(i), down.push(i);
85             if(pre.find(a[i]) != pre.end()) modify(1, 1, n, pre[a[i]] + 1, i - 1, -1);
86             else modify(1, 1, n, 1, i - 1, -1);
87             pre[a[i]] = i;
88             ans += query(1, 1, n, 1, i);
89         }
90         printf("Case #d: %lld\n", kase, ans);
91     }
92     return 0;
93 }

```



M.Acyclic Orientation



```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long LL;
4
5 // myfft
6 const int max0 = 262144 << 1;
7 const double PI = acos(-1);
8 int N, L, bitrev[max0 + 5];
9 struct cp {
10     double x, y;
11     cp() : x(0), y(0) {}
12     cp(const double &x, const double &y) : x(x), y(y) {}
13 } w[max0 + 5];
14 inline cp operator + (const cp &a, const cp &b) {return cp(a.x + b.x, a.y + b.y);}
15 inline cp operator - (const cp &a, const cp &b) {return cp(a.x - b.x, a.y - b.y);}
16 inline cp operator * (const cp &a, const cp &b) {return cp(a.x * b.x - a.y * b.y, a.x *
b.y + a.y * b.x);}
17 inline cp conj(const cp &a) {return cp(a.x, -a.y); }
18 void fft(cp *a, const int &n) {
19     for (int i = 0; i < n; ++i) if (i < bitrev[i]) swap(a[i], a[bitrev[i]]);
20     for (int i = 2, lyc = n >> 1; i <= n; i <= 1, lyc >= 1) {
21         for (int j = 0; j < n; j += i) {

```

```

22     cp *l = a + j, *r = a + j + (i >> 1), *p = w;
23     for (int k = 0; k < i >> 1; ++k) {
24         cp tmp = *r * *p;
25         *r = *l - tmp, *l = *l + tmp;
26         ++l, ++r, p += lyc;
27     }
28 }
29 }
30 }
31 inline void fft_prepare() {
32     for (int i = 0; i < N; ++i) bitrev[i] = bitrev[i >> 1] >> 1 | ((i & 1) << (L - 1));
33     for (int i = 0; i < N; ++i) w[i] = cp(cos(2 * PI * i / N), sin(2 * PI * i / N));
34 }
35 inline void conv(int *x, int *y, int *z, int mod) {
36     for (int i = 0; i < N; ++i) (x[i] += mod) %= mod, (y[i] += mod) %= mod;
37     static cp a[max0 + 5], b[max0 + 5];
38     static cp dfta[max0 + 5], dftb[max0 + 5], dftc[max0 + 5], dftd[max0 + 5];
39     for (int i = 0; i < N; ++i) a[i] = cp(x[i] & 32767, x[i] >> 15);
40     for (int i = 0; i < N; ++i) b[i] = cp(y[i] & 32767, y[i] >> 15);
41     fft(a, N), fft(b, N);
42     for (int i = 0; i < N; ++i) {
43         int j = (N - i) & (N - 1);
44         static cp da, db, dc, dd;
45         da = (a[i] + conj(a[j])) * cp(0.5, 0);
46         db = (a[i] - conj(a[j])) * cp(0, -0.5);
47         dc = (b[i] + conj(b[j])) * cp(0.5, 0);
48         dd = (b[i] - conj(b[j])) * cp(0, -0.5);
49         dfta[j] = da * dc;
50         dftb[j] = da * dd;
51         dftc[j] = db * dc;
52         dftd[j] = db * dd;
53     }
54     for (int i = 0; i < N; ++i) a[i] = dfta[i] + dftb[i] * cp(0, 1);
55     for (int i = 0; i < N; ++i) b[i] = dftc[i] + dftd[i] * cp(0, 1);
56     fft(a, N), fft(b, N);
57     for (int i = 0; i < N; ++i) {
58         int da = (LL) (a[i].x / N + 0.5) % mod;
59         int db = (LL) (a[i].y / N + 0.5) % mod;
60         int dc = (LL) (b[i].x / N + 0.5) % mod;
61         int dd = (LL) (b[i].y / N + 0.5) % mod;
62         z[i] = (da + ((LL) (db + dc) << 15) + ((LL) dd << 30)) % mod;
63     }
64 }
65
66 LL fac[max0], inv_fac[max0];
67 LL fp(LL a, LL b, LL mod) {
68     LL ret = 1LL;
69     while (b) {
70         if (b & 1) ret = ret * a % mod;
71         a = a * a % mod;
72         b >>= 1;
73     }
74     return ret;
75 }
76
77 LL S[max0], f[max0], pre[max0], suf[max0];
78 int main() {
79     int T;
80     scanf("%d", &T);
81     for (int kase = 1; kase <= T; ++kase) {
82         int n, m, mod;
83         scanf("%d %d %d", &n, &m, &mod);
84         fac[0] = 1;
85         for (int i = 1; i <= n + m; ++i) fac[i] = fac[i - 1] * i % mod;
86         inv_fac[0] = inv_fac[1] = 1;
87         for (int i = 2; i <= n + m; ++i) inv_fac[i] = (mod - (mod / i) * inv_fac[mod % i]
% mod) % mod;
88         for (int i = 3; i <= n + m; ++i) inv_fac[i] = inv_fac[i] * inv_fac[i - 1] % mod;
89         static int a[max0 + 5], b[max0 + 5], c[max0 + 5];
90         for (int i = 0; i <= n; ++i) a[i] = (i % 2 ? mod - 1 : 1) * inv_fac[i] % mod;
91         for (int i = 0; i <= n; ++i) b[i] = fp(i, n, mod) * inv_fac[i] % mod;
92         L = 0;
93         for (; (1 << L) < n + n + 1; ++L);

```

```
94     N = 1 << L;
95     for(int i = n + 1; i < N; ++i) a[i] = b[i] = 0;
96     fft_prepare();
97     conv(a, b, c, mod);
98     for(int i = 0; i <= n; ++i) S[i] = (c[i] + mod) * fac[i] % mod;
99     for(int i = 0; i <= n; ++i) a[i] = S[i] * inv_fac[i] % mod;
100    for(int i = 0; i <= n + m; ++i) b[i] = fp(i, m, mod) * inv_fac[i] % mod;
101    L = 0;
102    for (; (1 << L) < n + n + m + 1; ++L);
103    N = 1 << L;
104    for(int i = n + 1; i < N; ++i) a[i] = 0;
105    for(int i = n + m + 1; i < N; ++i) b[i] = 0;
106    fft_prepare();
107    conv(a, b, c, mod);
108    for(int i = 0; i <= n + m; ++i) f[i] = (c[i] + mod) * fac[i] % mod;
109    LL ans = 0;
110    pre[0] = suf[n + m + 2] = 1;
111    for(int i = 1; i <= n + m + 1; ++i) pre[i] = pre[i - 1] * (mod - i) % mod;
112    for(int i = n + m + 1; i >= 1; --i) suf[i] = suf[i + 1] * (mod - i) % mod;
113    for(int i = 1; i <= n + m + 1; ++i) {
114        LL x = pre[i - 1] * suf[i + 1] % mod * inv_fac[i-1] % mod * inv_fac[n + m + 1
- i] % mod;
115        if((n + m + 1 - i) % 2) x = x * (mod - 1) % mod;
116        ans = (ans + x * f[i - 1]) % mod;
117    }
118    if((n + m) % 2) ans = ans * (mod - 1) % mod;
119    printf("Case #%d: %lld\n", kase, ans);
120 }
121 return 0;
122 }
```

好文要顶

关注我

收藏该文

Aguin

关注 - 21

粉丝 - 35

+加关注

0

0

« 上一篇: [Facebook Hacker Cup 2018 Qualification Round](#)
» 下一篇: [2018 Multi-University Training Contest 1](#)

posted @ 2018-07-20 16:02 Aguin 阅读(324) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问 网站首页](#)。

- 【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码
- 【推荐】零基础轻松玩转云上产品，获赠礼加返百元大礼
- 【推荐】华为IoT平台开发者套餐9.9元起，购买即送免费课程

- 相关博文:**
- [The2018ACM-ICPCChineseCollegiateProgrammingContest](#)
 - [The13thChineseNortheastCollegiateProgrammingContestC.Line-lineIntersection](#)
 - [hdu 4497 GCD and LCM \(2013 ACM-ICPC吉林通化全国邀请赛——题目重现\)](#)
 - [HDU 4576 Robot \(概率题\)](#)
 - [大一下学期以来做题总结、](#)