矩阵快速幂

人员

左子毅、刘佳赫、杨洋 到课

上周作业检查

Begin: 2024-11-16 08:30 CST ☆ 2024-1116 ~ 1117 三队上课 **End:** 2024-12-28 00:30 CST

Elapsed: 6:23:48:59 Running **Remaining:** 34:16:11:00

Overview Problem Status Rank (6:23:48:52) Discuss Setting Clone Update Delete

Rank	Team	Score	Penalty	A 5 / 5	B 5 / 7	C 5 / 5	D 4 / 8
1	☆ ssine233 (?)	4	5771	**0:17:49**	**1:48:45**	**6:21:13**	3:15:04:11 (-2)
2	☆ hehe625 (刘佳赫)	4	7242	1:05:35:40	1:05:35:22	1:06:15:15	**1:06:36:13 (-2)**
3	☆ yujiahaoa (Pswd com...	3	28549	6:14:35:18		6:14:33:33	6:14:40:37
4	☆ syh123bc (_FL_)	2	197	1:28:05	1:49:50		
5	☆ syzxliangyuhan	2	223	1:52:09	1:51:39		
6	☆ James00123 (杨洋)	2	7033			2:10:36:13	2:10:37:19
7	☆ huhexuan	2	8375		5:22:21 (-2)	5:13:32:51	

作业

https://vjudge.net/contest/674606

课堂表现

这节课讲的矩阵快速幂，同学们课下要好好复习一下，同时也是熟悉一下封装的写法。

课堂内容

CF1561D1 Up the Strip (simplified version)

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int mod;
int f[maxn];

int main()
```

```
{
  int n; cin >> n >> mod;
  f[n] = 1;
  int sum = 0;
  for (int i = n; i >= 1; --i) {
    f[i] = (f[i] + sum) % mod;
    for (int l = 2, r; l <= i; l = r+1) {
      r = i / (i / l);
      int x = i / l;
      f[x] = (f[x] + LL(r-l+1)*f[i]) % mod;
    }
    sum = (sum + f[i]) % mod;
  }

  cout << f[1] << endl;
  return 0;
}
```

CF1561D2 Up the Strip

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 4e6 + 5;
int mod;
int f[maxn], suf[maxn];

int get_sum(int l, int r) { return (suf[l] - suf[r+1] + mod) % mod; }

int main()
{
  int n; cin >> n >> mod;
  f[n] = 1, suf[n] = 1;
  int sum = 1;
  for (int i = n-1; i >= 1; --i) {
    f[i] = (f[i] + sum) % mod;
    for (int j = 2; i*j <= n; ++j) {
      int l = i*j, r = min(n, (i+1)*j-1);
      f[i] = (f[i] + get_sum(l,r)) % mod;
    }
    sum = (sum + f[i]) % mod;
    suf[i] = (suf[i+1] + f[i]) % mod;
  }

  cout << f[1] << endl;
  return 0;
}
```

P3390 【模板】矩阵快速幂

```cpp
// 方法一
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 100 + 5;
const int mod = 1e9 + 7;

struct Matrix {
  int w[maxn][maxn];
  int n, m;

  Matrix() { n = m = 0, memset(w, 0, sizeof(w)); }
  Matrix(int x) { n = m = x, memset(w, 0, sizeof(w)); }
  Matrix(int x, int y) { n = x, m = y, memset(w, 0, sizeof(w)); }

  Matrix operator * (const Matrix& p) const {  // this(n,m) * p(n, m)
    int lenx = n, leny = m, lenz = p.m;
    Matrix c(lenx, lenz);

    for (int i = 1; i <= lenx; ++i) {
      for (int j = 1; j <= lenz; ++j) {
        for (int k = 1; k <= leny; ++k) {
          c.w[i][j] = (c.w[i][j] + (LL)w[i][k]*p.w[k][j]) % mod;
        }
      }
    }
    return c;
  }

  void build(int x) {
    n = m = x;
    memset(w, 0, sizeof(w));
    for (int i = 1; i <= x; ++i) w[i][i] = 1;
  }

  void read() {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= m; ++j) cin >> w[i][j];
    }
  }

  void print() {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= m; ++j) cout << w[i][j] << " ";
      cout << endl;
    }
  }
};
```

```
Matrix qmod(Matrix mtx, LL k) {
  Matrix res; res.build(mtx.n);
  while (k) {
    if (k&1) res = res * mtx;
    mtx = mtx * mtx;
    k >>= 1;
  }
  return res;
}

int main()
{
  int n; LL k; cin >> n >> k;
  Matrix mtx(n);
  mtx.read();
  Matrix res = qmod(mtx, k);
  res.print();
  return 0;
}
```

```
// 方法二
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 100 + 5;
const int mod = 1e9 + 7;

struct Matrix {
  int w[maxn][maxn];
  Matrix() { memset(w, 0, sizeof(w)); }
};

Matrix mul(Matrix a, Matrix b, int n) {
  Matrix res;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) {
      for (int k = 1; k <= n; ++k) {
        res.w[i][j] = (res.w[i][j] + (LL)a.w[i][k]*b.w[k][j]) % mod;
      }
    }
  }
  return res;
}

Matrix qmod(Matrix mtx, LL k, int n) {
  Matrix res;
  for (int i = 1; i <= n; ++i) res.w[i][i] = 1;

  while (k) {
```

```
    if (k & 1) res = mul(res, mtx, n);
    mtx = mul(mtx, mtx, n);
    k >>= 1;
  }
  return res;
}

int main()
{
  int n; LL k; cin >> n >> k;
  Matrix mtx;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) cin >> mtx.w[i][j];
  }

  Matrix res = qmod(mtx, k, n);
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) cout << res.w[i][j] << " ";
    cout << endl;
  }
  return 0;
}
```

P1962 斐波那契数列

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 100 + 5;
const int mod = 1e9 + 7;

struct Matrix {
  int w[maxn][maxn];
  int n, m;

  Matrix() { n = m = 0, memset(w, 0, sizeof(w)); }
  Matrix(int x) { n = m = x, memset(w, 0, sizeof(w)); }
  Matrix(int x, int y) { n = x, m = y, memset(w, 0, sizeof(w)); }

  Matrix operator * (const Matrix& p) const {  // this(n,m) * p(n, m)
    int lenx = n, leny = m, lenz = p.m;
    Matrix c(lenx, lenz);

    for (int i = 1; i <= lenx; ++i) {
      for (int j = 1; j <= lenz; ++j) {
        for (int k = 1; k <= leny; ++k) {
          c.w[i][j] = (c.w[i][j] + (LL)w[i][k]*p.w[k][j]) % mod;
        }
      }
```

```
    }
    return c;
  }

  void build(int x) {
    n = m = x;
    memset(w, 0, sizeof(w));
    for (int i = 1; i <= x; ++i) w[i][i] = 1;
  }

  void read() {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= m; ++j) cin >> w[i][j];
    }
  }

  void print() {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= m; ++j) cout << w[i][j] << " ";
      cout << endl;
    }
  }
};

Matrix qmod(Matrix mtx, LL k) {
  Matrix res; res.build(mtx.n);
  while (k) {
    if (k&1) res = res * mtx;
    mtx = mtx * mtx;
    k >>= 1;
  }
  return res;
}

int main()
{
  LL n; cin >> n;
  if (n <= 2) { cout << 1 << endl; return 0; }

  Matrix mtx(2, 2); mtx.w[1][1] = mtx.w[1][2] = mtx.w[2][1] = 1;
  mtx = qmod(mtx, n-2);

  Matrix a(1, 2); a.w[1][1] = a.w[1][2] = 1;
  a = a * mtx;

  cout << a.w[1][1] << endl;
  return 0;
}
```