Suhendry's Blog

WHEN IN DOUBT, DO MATH.

Jan
10
2010

Matrix Multiplication

Algorithm, Programming

Add comments

Matrix multiplication could be useful to solve some problems. For example, what is the last digit of 1,000,000,000th fibonacci? Calculating fibonacci number using dynamic programming will need O(n) time complexity, we need a faster one to solve this problem.

Let A be a 1 x 2 matrix which consist of f_0 and f_1 (ie. 0 and 1) and B be a 2 x 2 matrix which consist of {(0,1),(1,1)}. If we multiply A and B, we will get a 1 x 2 matrix C which consist of f_1 and f_0+f_1 (= f_2). If we multiply C with B, then we will get a matrix D which consist of f_2 and f_1+f_2 (= f_3). By doing this, we can obtain f_n by multiplying A with B for n-1 times.

$$\begin{pmatrix} f_0 & f_1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} f_1 & f_2 \end{pmatrix}$$

$$\begin{pmatrix} f_1 & f_2 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} f_2 & f_3 \end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix} f_0 & f_1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \cdots \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} f_n & f_{n+1} \end{pmatrix}$$

$$\begin{pmatrix} f_0 & f_1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} f_n & f_{n+1} \end{pmatrix}$$

Matrix multiplication is associative, so we can compute f_n by multiplying A with B to the power of n-1. We know that a^n can be computed in O(lg n) and multiplying two matrixes needs O(s^3) where s is the size of the matrix, hence the total time complexity would be O(s^3 . lg n).

Here is an example code of computing the last digit of nth fibonacci number.

```
typedef vector<vector<int> > vvi;
const int mod = 10;

vvi mul(const vvi &a, const vvi &b) {
  vvi ret(a.size(),b[0].size());
  REP(i,a.size()) REP(j,b[i].size())
    REP(k,a[i].size()) ret[i][j] = (ret[i][j] + a[i][k] * b[k][j]) % mod;
  return ret;
}

vvi power(const vvi &a, int p) {
  if ( p == 1 ) return a;
  vvi x = power(a,p/2);
  if ( p % 2 == 1 ) return mul(a,mul(x,x));
  return mul(x,x);
}

int main()
{
  vvi a(1,2), b(2,2);
  a[0][1] = 0, a[0][1] = 1;
  b[0][0] = 0, b[0][1] = b[1][0] = b[1][1] = 1;

  int n;
  scanf( "%d", &n );
  printf( "%d\n", n == 0 ? 0 : mul(a,power(b,n))[0][0] );

  return 0;
}
```

Now let's consider another problem. Given a graph G of k node, how many distinct path of length n are there that starts from node 1? This problem could also be solved by matrix multiplication.

Categories

Blogroll

Andrian Kurniady
Brian Marshal
Eko Wibowo
Felix Halim
Fiona Liausvia
Lego Haryanto
Risan
Timotius Sakti
Yen Lina Prasetyo

Link

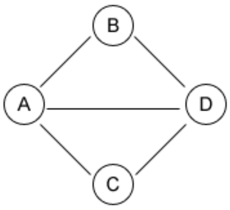
PKU Online Judge
Project Euler
Sphere Online Judge
TJU Online Judge
TopCoder Algorithm
USACO Training Gate
UVA Online Judge
ZJU Online Judge

Panda

White Tiger

Go

Let A be a 1 x k matrix which consist of the number of path that end in each node, for the initial case of course it will be (1, 0, …, 0). Let B the adjacency matrix of graph G. If we multiply A with B, we will get a matrix C which consist of the number of path of length-1 that end in each node. If we multiply C with B again, we will get a matrix D which consist of the number of path of length-2 that end in each node, and so on.



$$\begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix} \times \begin{pmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1\\ 1 & 1 & 1 & 0\end{pmatrix} = \begin{pmatrix}0 & 1 & 1 & 1\end{pmatrix}$$

$$\begin{pmatrix}0 & 1 & 1 & 1\end{pmatrix} \times \begin{pmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1\\ 1 & 1 & 1 & 0\end{pmatrix} = \begin{pmatrix}3 & 1 & 1 & 2\end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix}A_0 & B_0 & C_0 & D_0\end{pmatrix} \times \begin{pmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1\\ 1 & 1 & 1 & 0\end{pmatrix}^n = \begin{pmatrix}A_n & B_n & C_n & D_n\end{pmatrix}$$

There are 3 + 1 + 1 + 2 = 7 paths of length 2 in the graph above:
A – B – A
A – C – A
A – D – A
A – D – B
A – D – C
A – B – D
A – C – D

Exercises:
TJU 2169 – Number Sequence
TJU 2871 – Magic Bean
UVA 11149 – The Power of Matrix
SPOJ SEQ – Recursive Sequence
Live Archieve 4332 – Blocks for kids

Posted by suhendry at 8:53 pm

3 Responses to "Matrix Multiplication"

1. **Kurniady** says:
   January 11, 2010 at 5:06 pm

   For exercise:

   let's define F(i) = F(i-1) + F(i-2) + X where X is a constant.

   Use the matrix way to calculate the last digit of F(1,000,000,000)

   😉

   -Kurniady

   Reply

2. **mahli** says:
   January 12, 2010 at 7:52 pm

   problem H?

   nice tutorial, btw.

   Reply

3. **Lego Haryanto** says:
   March 28, 2010 at 12:19 pm

   Have you tried Project Euler 237, … this is also another use of some matrix multiplication stuff, but tougher.

   Reply

Leave a Reply

Your Comment

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <s> <strike> <strong>`

Name (required)

E-mail (required)

URI

Submit Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.

ICPC Indonesia National Contest 2009 ACM-ICPC 2009 Jakarta – Problem Set & Analysis