Fast Matrix Exponentiation – Introduction

(Prepared by the UCF Programming Team Coaches for the Developmental Teams)

First, a quick refresher for matrix multiplication:

$$\underset{p \times q}{\overset{\text{Matrix A}}{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1q} \\ a_{21} & a_{22} & \dots & a_{2q} \\ \dots & & & \\ a_{p1} & a_{p2} & \dots & a_{pq} \end{pmatrix}}} \underset{q \times t}{\overset{\text{Matrix B}}{\begin{pmatrix} b_{11} & b_{12} & \dots & b_{1t} \\ b_{21} & b_{22} & \dots & b_{2t} \\ \dots & & & \\ b_{q1} & b_{q2} & \dots & b_{qt} \end{pmatrix}}} \quad \underset{p \times t}{\overset{\text{Matrix C}}{\begin{pmatrix} c_{11} & c_{12} & \dots & c_{1t} \\ c_{21} & c_{22} & \dots & c_{2t} \\ \dots & & & \\ c_{p1} & c_{p2} & \dots & c_{pt} \end{pmatrix}}}$$

```
for i = 1 to number_of_rows[A]
     for j = 1 to number_of_columns[B]

          C[i][j] = 0
          for k = 1 to number_of_columns[A]
               C[i][j] = C[i][j] + (A[i][k] * B[k][j])
          end for

     end for
end for
```

Now, on to the main topic:
Let's assume we want to compute $base^{exp}$ using multiplications, e.g., we want to compute 24^{500}.
Let's start with the straightforward solutions:

Iterative:

```
int power(int base, int exp)
{
     int k, result;
     result = 1;
     for (k = 1;  k <= exp;  ++k)
          result = result * base;

     return(result);

}/* end of power() */
```

Recursive:

```
int power(int base, int exp)
{
     if (exp == 0)
          return(1);

     return(base * power(base, exp - 1));

}/* end of power() */
```

These solutions require O(*exp*) multiplications.

But we should be able to do better! We can compute $base^{exp/2}$ (let's call this intermediate result *half*), then the final result is "*half* * *half*" if *exp* is even or the final result is "*base* * *half* * *half*" if exp is odd. Using our earlier example:

To compute 24^{500}, if *half* = 24^{250}, then 24^{500} = *half* * *half*.

To compute 24^{250}, if *half* = 24^{125}, then 24^{250} = *half* * *half*.

To compute 24^{125}, if *half* = 24^{62}, then 24^{125} = 24 * *half* * *half*.

Here is the code:

```
int power(int base, int exp)
{
     if (exp == 0)
          return(1);

     int half = power(base, exp / 2);
     if (exp % 2 == 0)
          return(half * half)
     else
          return(base * half * half);

}/* end of power() */
```

Since this solution is cutting the size in half each time, it requires O(log *exp*) multiplications.

This concept can be used in matrix exponentiation as well, i.e., when we need to multiply a matrix with itself several times, i.e., we want to compute a matrix raised to an exponent (program/code on Google Drive).

This efficient way of matrix exponentiation can be used to compute values in a recurrence relation.

Example: Fibonacci Numbers

$F_0 = 0$
$F_1 = 1$
$F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$

Computing F_n when *n* is large takes a long time. We can use the above concept (fast matrix exponentiation) to do this faster:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_1 + F_0 \\ F_1 + 0 \end{bmatrix} = \begin{bmatrix} F_2 \\ F_1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_2 \\ F_1 \end{bmatrix} = \begin{bmatrix} F_2 + F_1 \\ F_2 + 0 \end{bmatrix} = \begin{bmatrix} F_3 \\ F_2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} F_3 + F_2 \\ F_3 + 0 \end{bmatrix} = \begin{bmatrix} F_4 \\ F_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix} = \begin{bmatrix} F_{n-1} + F_{n-2} \\ F_{n-1} + 0 \end{bmatrix} = \begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix}$$

Another way of looking at the Fibonacci Sequence:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^1 = \begin{bmatrix} F_2 & F_1 \\ F_1 & F_0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} F_3 & F_2 \\ F_2 & F_1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^3 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} F_4 & F_3 \\ F_3 & F_2 \end{bmatrix}$$

$$\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}^4 = \begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix} = \begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}3 & 2\\2 & 1\end{bmatrix} = \begin{bmatrix}5 & 3\\3 & 2\end{bmatrix} = \begin{bmatrix}F_5 & F_4\\F_4 & F_3\end{bmatrix}$$

$$\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}^n = \begin{bmatrix}F_{n+1} & F_n\\F_n & F_{n-1}\end{bmatrix}$$

So, we can use matrix exponentiation to compute F_n efficiently even when *n* is large.

To generalize this process one step further (i.e., other recurrence relations):

If $F_n = a.F_{n-1} + b.F_{n-2}$ for $n \geq 2$

then

$$\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_1\\F_0\end{bmatrix} = \begin{bmatrix}a.F_1 + b.F_0\\F_1 + 0\end{bmatrix} = \begin{bmatrix}F_2\\F_1\end{bmatrix}$$

$$\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_1\\F_0\end{bmatrix} = \begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_2\\F_1\end{bmatrix} = \begin{bmatrix}a.F_2 + b.F_1\\F_2 + 0\end{bmatrix} = \begin{bmatrix}F_3\\F_2\end{bmatrix}$$

$$\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_1\\F_0\end{bmatrix} = \begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_3\\F_2\end{bmatrix} = \begin{bmatrix}a.F_3 + b.F_2\\F_3 + 0\end{bmatrix} = \begin{bmatrix}F_4\\F_3\end{bmatrix}$$

$$\begin{bmatrix}a & b\\1 & 0\end{bmatrix}^n\begin{bmatrix}F_1\\F_0\end{bmatrix} = \begin{bmatrix}F_{n+1}\\F_n\end{bmatrix}$$

$$\begin{bmatrix}a & b\\1 & 0\end{bmatrix}\begin{bmatrix}F_{n-1}\\F_{n-2}\end{bmatrix} = \begin{bmatrix}a.F_{n-1} + b.F_{n-2}\\F_{n-1} + 0\end{bmatrix} = \begin{bmatrix}F_n\\F_{n-1}\end{bmatrix}$$

To generalize the process one step further (i.e., other recurrence relations):

If $F_n = a.F_{n-1} + b.F_{n-2} + c.F_{n-3}$ for $n \geq 3$

then we need to derive:
$Matrix_1 * Matrix_2 = Matrix_3$

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_{n-2} \\ F_{n-3} \end{bmatrix} = \begin{bmatrix} a.F_{n-1} + b.F_{n-2} + c.F_{n-3} \\ F_{n-1} + 0 + 0 \\ 0 + F_{n-2} + 0 \end{bmatrix} = \begin{bmatrix} F_n \\ F_{n-1} \\ F_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_2 \\ F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} a.F_2 + b.F_1 + c.F_0 \\ F_2 + 0 + 0 \\ 0 + F_1 + 0 \end{bmatrix} = \begin{bmatrix} F_3 \\ F_2 \\ F_1 \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_2 \\ F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_3 \\ F_2 \\ F_1 \end{bmatrix} = \begin{bmatrix} a.F_3 + b.F_2 + c.F_1 \\ F_3 + 0 + 0 \\ 0 + F_2 + 0 \end{bmatrix} = \begin{bmatrix} F_4 \\ F_3 \\ F_2 \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_2 \\ F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_4 \\ F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} a.F_4 + b.F_3 + c.F_2 \\ F_4 + 0 + 0 \\ 0 + F_3 + 0 \end{bmatrix} = \begin{bmatrix} F_5 \\ F_4 \\ F_3 \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^n \begin{bmatrix} F_2 \\ F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_{n+2} \\ F_{n+1} \\ F_n \end{bmatrix}$$