

I, ME AND MYSELF !!!

SATURDAY, NOVEMBER 20, 2010

Matrix Exponentiation

Introduction:

Don't be confused with the title, this article has nothing to do with "how to calculate an exponent of a given matrix", rather it will discuss on how to use this technique to solve a specific class of problems.

Sometimes we face some problems, where, we can easily derive a recursive relation (mostly suitable for dynamic programming approach), but the given constraints make us about to cry, there comes the matrix exponentiation idea. The situation can be made more clear with the following example:

Let, a problem says: *find $f(n)$: n 'th Fibonacci number*. When n is sufficiently small, we can solve the problem with a simple recursion, $f(n) = f(n-1) + f(n-2)$, or, we can follow dynamic programming approach to avoid the calculation of same function over and over again. But, what will you do if the problem says, *given $0 < n < 1000000000$, find $f(n) \% 999983$* ? No doubt dynamic programming will fail!

We'll develop the idea on how and why these types of problems could be solved by matrix exponentiation later, first lets see how matrix exponentiation can help is to represent recurrence relations.

Prerequisite:

Before continuing, you need to know:

- Given two matrices, how to find their product, or given the product matrix of two matrices, and one of them, how to find the other matrix.
- Given a matrix of size $d \times d$, how to find its n 'th power in $O(d^3 \log(n))$.

Patterns:

What we need first, is a recursive relation, and what we want to do, is to find a matrix M which can lead us to the desired state from a set of already known states. Let, we know k states of a given recurrence relation, and want to find the $(k+1)$ th state. Let M be a $k \times k$ matrix, and we build a matrix $A:[k \times 1]$ matrix from the known states of the recurrence relation, now we want to get a matrix $B:[k \times 1]$ which will represent the set of next states, i.e. $M \times A = B$, as shown below:

$$\begin{array}{c}
 \begin{array}{c|c|c}
 f(n) & & f(n+1) \\
 f(n-1) & & f(n) \\
 f(n-2) & & f(n-1) \\
 \dots & & \dots \\
 f(n-k) & & f(n-k+1)
 \end{array} \\
 \times \begin{array}{c} \\ \\ \\ \\ \\
 \end{array} = \begin{array}{c} \\ \\ \\ \\ \\
 \end{array}
 \end{array}$$

So, if we can design M accordingly, job's done!, the matrix will then be used to represent the recurrence relation.

• Type 1:

Lets start by the simplest one, $f(n) = f(n-1) + f(n-2)$.

So, $f(n+1) = f(n) + f(n-1)$

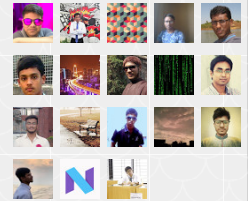
Let, we know, $f(n)$ and $f(n-1)$; we want to get $f(n+1)$

From the above situation, matrix A and B can be formed as shown below:

Matrix A

Matrix B

Followers (557) [Next](#)



[Follow](#)

SUBSCRIBE

[Posts](#)

[Comments](#)

BLOG HITS



6 9 5 5 3 2

BLOG ARCHIVE

- [2015](#) (4)
- [2014](#) (6)
- [2013](#) (19)
- [2012](#) (14)
- [2011](#) (15)
- ▼ [2010](#) (33)
 - [December](#) (7)
 - ▼ [November](#) (3)
 - [CRC - 32 \(IEEE\)](#)
 - [Matrix Exponentiation](#)
 - [Merge Sort Improvement?](#)
 - [September](#) (1)
 - [August](#) (3)
 - [June](#) (2)
 - [May](#) (5)
 - [March](#) (5)
 - [February](#) (4)
 - [January](#) (3)
- [2009](#) (27)

$$\begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} \quad \begin{array}{c|c} f(n+1) & \\ \hline f(n) & \end{array}$$

[Note: matrix A will be always designed in such a way that, every state on which $f(n+1)$ depends, will be present]

So, now, we need to design a 2×2 matrix M such that, it satisfies $M \times A = B$ as stated above.

The first element of B is $f(n+1)$ which is actually $f(n) + f(n-1)$. To get this, from matrix A , we need, 1 $f(n)$ and 1 $f(n-1)$. So, the 1st row of M will be $[1 \ 1]$.

$$\begin{array}{c|c} 1 & 1 \\ \hline \text{-----} & \end{array} \times \begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} = \begin{array}{c|c} f(n+1) & \\ \hline f(n) & \end{array}$$

[Note: ----- means, we are not concerned about this value]

Similarly, 2nd item of B is $f(n)$ which we can get by simply taking 1 $f(n)$ from A . So, the 2nd row of M is $[1 \ 0]$.

$$\begin{array}{c|c} \text{-----} & \\ \hline 1 & 0 \end{array} \times \begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} = \begin{array}{c|c} \text{-----} & \\ \hline f(n) & \end{array}$$

[I hope you know how a matrix multiplication is done and how the values are assigned!]

Thus we get the desired 2×2 matrix M :

$$\begin{array}{c|c} 1 & 1 \\ \hline 1 & 0 \end{array} \times \begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} = \begin{array}{c|c} f(n+1) & \\ \hline f(n) & \end{array}$$

If you are confused about how the above matrix is calculated, you might try doing it this way:

We know, the multiplication of an $n \times n$ matrix M with an $n \times 1$ matrix A will generate an $n \times 1$ matrix B , i.e. $M \times A = B$. The k 'th element in the product matrix B is the product of k 'th row of the $n \times n$ matrix M with the $n \times 1$ matrix A in the left side.

In the above example, the 1st element in B is $f(n+1) = f(n) + f(n-1)$. So, it's the product of 1st row of matrix M and matrix A . Let, the first row of M is $[x \ y]$. So, according to matrix multiplication,

$$x * f(n) + y * f(n-1) = f(n+1) = f(n) + f(n-1) \\ \Rightarrow x = 1, y = 1$$

Thus we can find the first row of matrix M is $[1 \ 1]$. Similarly, let, the 2nd row of matrix M is $[x \ y]$, and according to matrix multiplication:

$$x * f(n) + y * f(n-1) = f(n) \\ \Rightarrow x = 1, y = 0$$

Thus we get the second row of M is $[1 \ 0]$.

• Type 2:

Now, we make it a bit complex: find $f(n) = a * f(n-1) + b * f(n-2)$, where a, b are some constants.

This tells us, $f(n+1) = a * f(n) + b * f(n-1)$.

By this far, this should be clear that the dimension of the matrices will be equal to the number of dependencies, i.e. in this particular example, again 2. So, for A and B , we can build two matrices of size 2×1 :

Matrix A

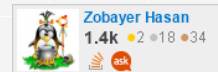
Matrix B

$$\begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} \quad \begin{array}{c|c} f(n+1) & \\ \hline f(n) & \end{array}$$

Now for $f(n+1) = a * f(n) + b * f(n-1)$, we need $[a \ b]$ in the first row of objective matrix M instead of $[1 \ 1]$ from the previous example. Because, now we need a of $f(n)$'s and b of $f(n-1)$'s.

$$\begin{array}{c|c} a & b \\ \hline \text{-----} & \end{array} \times \begin{array}{c|c} f(n) & \\ \hline f(n-1) & \end{array} = \begin{array}{c|c} f(n+1) & \\ \hline f(n) & \end{array}$$

STACK OVERFLOW



And, for the 2nd item in B i.e. $f(n)$, we already have that in matrix A , so we just take that, which leads, the 2nd row of the matrix M will be $[1 \ 0]$ as the previous one.

$$\begin{array}{c|c|c|c|c|c|c} & \text{-----} & | & x & | & f(n) & | & = & | & \text{-----} & | \\ & 1 & 0 & | & & f(n-1) & | & & & f(n) & | \end{array}$$

So, this time we get:

$$\begin{array}{c|c|c|c|c|c|c} & a & b & | & x & | & f(n) & | & = & | & f(n+1) & | \\ & 1 & 0 & | & & f(n-1) & | & & & f(n) & | \end{array}$$

Pretty simple as the previous one...

• Type 3:

We've already grown much older, now let's face a bit complex relation: find $f(n) = a * f(n-1) + c * f(n-3)$.
Ooops! a few minutes ago, all we saw were contiguous states, but here, the state $f(n-2)$ is missing. Now? what to do?

Actually, this is not a problem anymore, we can convert the relation as follows: $f(n) = a * f(n-1) + 0 * f(n-2) + c * f(n-3)$, deducing $f(n+1) = a * f(n) + 0 * f(n-1) + c * f(n-2)$. Now, we see that, this is actually a form described in Type 2. So, here, the objective matrix M will be 3×3 , and the elements are:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} & a & 0 & c & | & & f(n) & | & & f(n+1) & | \\ & 1 & 0 & 0 & | & x & f(n-1) & | & = & & f(n) & | \\ & 0 & 1 & 0 & | & & f(n-2) & | & & f(n-1) & | \end{array}$$

These are calculated in the same way as Type 2. [Note, if you find it difficult, try on pen and paper!]

• Type 4:

Life is getting complex as hell, and Mr. problem now asks you to find $f(n) = f(n-1) + f(n-2) + c$ where c is any constant.

Now, this is a new one and all we have seen in past, after the multiplication, each state in A transforms to its next state in B .

$$f(n) = f(n-1) + f(n-2) + c$$

$$f(n+1) = f(n) + f(n-1) + c$$

$$f(n+2) = f(n+1) + f(n) + c$$

..... so on

So, normally we can't get it through the previous fashions. But, how about now we add c as a state?

$$\begin{array}{c|c|c|c|c|c|c} & f(n) & | & & f(n+1) & | \\ M \times & f(n-1) & | & = & f(n) & | \\ & c & | & & c & | \end{array}$$

Now, it's not much hard to design M according to the previous fashion. Here it is done, but don't forget to verify on yours:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} & 1 & 1 & 1 & | & & f(n) & | & & f(n+1) & | \\ & 1 & 0 & 0 & | & x & f(n-1) & | & = & & f(n) & | \\ & 0 & 0 & 1 & | & & c & | & & c & | \end{array}$$

• Type 5:

Lets put it altogether: find matrix suitable for $f(n) = a * f(n-1) + c * f(n-3) + d * f(n-4) + e$.

I would leave it as an exercise to reader. The final matrix is given here, check if it matches with your solution. Also find matrix A and B .

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} & a & 0 & c & d & 1 & | \\ & 1 & 0 & 0 & 0 & 0 & | \\ & 0 & 1 & 0 & 0 & 0 & | \\ & 0 & 0 & 1 & 0 & 0 & | \\ & 0 & 0 & 0 & 0 & 1 & | \end{array}$$

[Note: you may take a look back to Type 3 and 4]

- **Type 6:**

Sometimes, a recurrence is given like this:

```
f(n) = if n is odd, f(n-1) else, f(n-2)
In short:
f(n) = (n&1) * f(n-1) + (! (n&1)) * f(n-2)
```

Here, we can just split the functions in the basis of odd even and keep 2 different matrix for both of them and calculate separately. Actually, there might appear many different patterns, but these are the basic patterns.

- **Type 7:**

Sometimes we may need to maintain more than one recurrence, where they are interrelated. For example, let a recurrence relation be:

$g(n) = 2g(n-1) + 2g(n-2) + f(n)$, where, $f(n) = 2f(n-1) + 2f(n-2)$. Here, recurrence $g(n)$ is dependent upon $f(n)$ and the can be calculated in the same matrix but of increased dimensions. Lets design the matrices A , B then we'll try to find matrix M .

Matrix A

$$\begin{bmatrix} g(n) \\ g(n-1) \\ f(n+1) \\ f(n) \end{bmatrix}$$

Matrix B

$$\begin{bmatrix} g(n+1) \\ g(n) \\ f(n+2) \\ f(n+1) \end{bmatrix}$$

Here, $g(n+1) = 2g(n) + 2g(n-1) + f(n+1)$ and $f(n+2) = 2f(n+1) + 2f(n)$.

Now, using the above process, we can generate the objective matrix M as follows:

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

So, these are the basic categories of recurrence relations which are used to be solved by this simple technique.

Analysis:

Now that we have seen how matrix multiplication can be used to maintain recurrence relations, we are back to our first question, how this helps us in solving recurrences on a huge range.

Recall the recurrence $f(n) = f(n-1) + f(n-2)$.

We already know that:

$$M \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix} \dots\dots\dots(1)$$

How about we multiply M multiple times? Like this:

$$M \times M \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

Replacing from (1):

$$M \times M \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = M \times \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix} = \begin{bmatrix} f(n+2) \\ f(n+1) \end{bmatrix}$$

So, we finally get:

$$M^2 \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+2) \\ f(n+1) \end{bmatrix}$$

Similarly we can show:

$$M^3 \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+3) \\ f(n+2) \end{bmatrix}$$

$$M^4 \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+4) \\ f(n+3) \end{bmatrix}$$

$$\dots\dots\dots$$

$$M^k \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+k) \\ f(n+k-1) \end{bmatrix}$$

Thus we can get any state $f(n)$ by simply raising the power of objective matrix M to $n-1$ in $O(d^3 \log(n))$, where d is the dimension of square matrix M . So, even if $n = 1000000000$, still this can be calculated pretty easily as long as d^3 is sufficiently small.

Related problems:

- UVa 10229 : Modular Fibonacci
- UVa 10870 : Recurrences
- UVa 11651 : Krypton Number System
- UVa 10754 : Fantastic Sequence
- UVa 11551 : Experienced Endeavour

Regards, Zobayer Hasan.

Posted by [Unknown](#) at [2:12 PM](#)

96 comments:

Anonymous November 20, 2010 at 3:17 PM

Thank you, Zobayer.
It's worth reading :)

[Reply](#)

Anonymous November 26, 2010 at 9:45 AM

Hey Zobayer. Do have any idea to solve that:
we consider $F(n)$ function determined by following term
if n is quadratic value then $F(n)=0$
otherwise $f(n) = [1 / \{\text{sqrt}(n)\}]$
So what is $S = F(1)+F(2)+...+F(N^2)$.
I think there is no formula exists to solve it. But How do i solve it easier??

[Reply](#)

 **Zobayer Hasan** November 30, 2010 at 3:59 AM

I don't think I can solve this type using matrix exponentiation, I can go only for linear ones. I don't think quadratic equations can be solved.

But, still this could be solved, like, sometimes approximation formulas can be found, or, you can also try divide and conquer / dp techniques. Might help.

[Reply](#)



Sabbir Yousuf Sanny December 21, 2010 at 4:45 AM

Really nice tutorial! Helped me to solve this problem on latest Codechef - <http://www.codechef.com/COOK05/problems/SEEDS/>

Keep it up!

[Reply](#)

Sankalp January 9, 2011 at 5:18 PM

Hey Zobayer!

I have a problem where the recursive relation is like this:

$$f(n, h, k) = f(n-1, h-1, k+1) + f(n-1, h, k-1) + f(n-1, h+1, k)+1$$

do you have any idea how to solve this by matrix exponentiation..?

[Reply](#)



Zobayer Hasan January 9, 2011 at 10:25 PM

Is it not possible with dp? I don't think it's possible with matrix exponentiation as all of the parameters are not shifting at a same order. I am sorry if I am wrong, actually I don't know the answer of your question :(

[Reply](#)



Aakash Johari (IIITA) February 2, 2011 at 6:25 PM

Hi, Zobayar. Can you explain type-7 matrix A and B. According to me it should be

$$| g(n) |$$

$$A = | g(n-1) |$$

$$| f(n) |$$

$$| f(n-1) |$$

$$B = | g(n+1) |$$

$$| g(n) |$$

$$| f(n+1) |$$

$$| f(n) |$$

[Reply](#)



Zobayer Hasan February 2, 2011 at 10:13 PM

@Aakash, if you want to find out $g(n+1)$ then you will be needing:

$$g(n+1) = 2g(n) + 2g(n-1) + f(n+1)$$

So, you must have $f(n+1)$ on the left side. Hope you get it.

[Reply](#)



Radhakrishnan February 16, 2011 at 1:25 AM

This comment has been removed by the author.

[Reply](#)



Zobayer Hasan February 16, 2011 at 1:55 AM

nice question. look carefully, you can always do something like this, say for $n = 3$

$$1 + A + A^2 + A^3$$

$$= (1 + A) + A^2(1 + A)$$

so divide and conquer is possible.

I have a post related to this, which u may wish to see.

<http://zobayer.blogspot.com/2009/12/power-series-evaluation.html>

thanks for commenting.

[Reply](#)

Anonymous April 16, 2011 at 7:31 PM

Thanks for the informative post!

It would be more helpful, if you could point out/give links to resources about finding n 'th power of matrix in $O(d^3 * \log(n))$.

[Reply](#)



Zobayer Hasan April 16, 2011 at 10:05 PM

I think, you can find b^n in $\log(n)$ right?

a recursive algorithm might look like this:

$f(b, n)$:

----If $n == 0$ return 1

----If n is odd return $f(b, n-1) * b$;

----else return $\text{square}(f(b, n/2))$

See, you are multiplying b here which is definitely $O(1)$, however, for matrix, b will be a $d \times d$ matrix, and to multiply two $d \times d$ matrix, you need $O(d^3)$. So the actual complexity is $O(d^3 \lg(n))$

[Reply](#)

Anonymous May 27, 2011 at 2:23 AM

hey can you tell about uva 11651 I am not able to figure it out

[Reply](#)

Anonymous May 27, 2011 at 6:13 PM

excellent one.....worth reading

[Reply](#)

Voltoorb October 8, 2011 at 3:24 PM

I tried the HDU 2802 problem using matrix exponentiation. I got TLE, so tried storing the matrices for A, A^2 , A^4 ,... to quicken the calculation of A^n . But still got TLE!

Can this be solved using matrix exponentiation for sure? Because there is this Chinese blog, which tells there is some cycle form - <http://hi.baidu.com/xiinho/blog/item/1307e6156a9be10d4b90a741.html>

[Reply](#)



Zobayer Hasan October 9, 2011 at 10:11 PM

I could not do it with matrix expo either, but I have listed it here because I have found this problem in the list of matrix expo problems somewhere, can't remember now where, many days passed. Sorry!

[Reply](#)

Anonymous December 22, 2011 at 6:16 PM

$f(n) = n * f(n-1) + f(n-2)$

Can this be solved using matrix exponentiation ?

[Reply](#)

Anonymous March 12, 2012 at 5:15 PM

Hi Zobayer,

Can you show how to find the result for

$f(n) = a * f(n-1) + b$?

Thanks :)

[Reply](#)

▼ Replies



Zobayer Hasan March 12, 2012 at 9:04 PM

This is quite simple, I think you can try the following relation
 $\{\{a, 1\}, \{0, 1\}\} * \{\{f(n), b\}\} = \{f(n+1), b\}$

[Reply](#)

Anonymous June 4, 2012 at 2:25 PM

Hmm...the new theme is cool :-), nice article btw. Although I knew about matrix exponentiation before but learned a couple of new things here. Very well explained too.

[Reply](#)

▼ Replies



Zobayer Hasan June 4, 2012 at 9:51 PM

Thank you :)

[Reply](#)

Anonymous June 5, 2012 at 11:28 AM

Can we do it for recurrence having constant as a function of n.
Like for example $f(n) = f(n-1) + f(n-2) + n$?

[Reply](#)

▼ Replies



Zobayer Hasan June 5, 2012 at 5:50 PM

I don't think so, or to be precise, I can't.

Nobody November 9, 2013 at 4:07 PM

Yes, we can do it. Just define $g(n) = n \Rightarrow g(n) = g(n-1) + 1, g(1) = 1$.
Then, $f(n+1) = f(n) + f(n-1) + g(n+1) = f(n) + f(n-1) + (n+1)$.
This is of type 7 as given in the article. We can solve it as:

$$\begin{bmatrix} f(n+1) \\ f(n) \\ g(n+2) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f(n) \\ f(n-1) \\ g(n+1) \\ 1 \end{bmatrix}$$

We need to find out $\{ \{ f(n+1) \}, \{ f(n) \}, \{ g(n+2) \}, \{ 1 \} \}$.

We know $\{ \{ f(n) \}, \{ f(n-1) \}, \{ g(n+1) \}, \{ 1 \} \}$.

The matrix is

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And thank you Zobayer for this wonderful article. It helped me a lot.



Zobayer Hasan April 23, 2015 at 10:10 AM

@Nobody, clever thinking, would you mind if I add this to the article? I wish I knew your email address / website or something.

[Reply](#)

Muhammed Hedayet June 7, 2012 at 1:28 AM

Cool. I know Matrix exponentiation technique and have solved some problems using this technique, but nice organized article. This article let me think in a new fashion. Thanks. Want more. . . :-)

[Reply](#)

▼ Replies



Zobayer Hasan June 7, 2012 at 2:03 AM

Thanks :)

[Reply](#)

Anonymous July 3, 2012 at 12:18 PM

Very nice post, keep up the great work.

[Reply](#)



Shahin ShamS July 3, 2012 at 6:31 PM

I didn't know about the topic before read the article. Many many thanks for this.

[Reply](#)



Shahin ShamS July 3, 2012 at 6:33 PM

Many many thanks to your for this article. Because the topic is new for me.... ans thanks again.

[Reply](#)



Shahin ShamS July 3, 2012 at 6:36 PM

Many many thanks to you for this article. Because the topic is new for me..... thanks again.

[Reply](#)



sahil sharma July 7, 2012 at 11:07 AM

good work dude..really nice

[Reply](#)



riazur July 8, 2012 at 4:07 AM

Thanks Vaia.It's really helpful.Thanks a lot :)

[Reply](#)

sameer July 17, 2012 at 12:42 PM

Thanks for sharing.
Keep up the good work !!!

[Reply](#)

Anonymous September 5, 2012 at 5:37 AM

Can we do it if we have a n on the right side?

For example:

$f(n) = a*f(n-1) - n$?

[Reply](#)

▼ [Replies](#)

Nobody November 9, 2013 at 4:11 PM

Yes we can. Just define $g(n) = n \Rightarrow g(n) = g(n-1) + 1, g(1) = 1$.
Then $f(n) = a*f(n-1) - g(n)$. This is of type 7 as given in the article.
We need to find out the 1 column matrix:

$f(n+1)$

$g(n+2)$

1

We know:

$f(n)$

$g(n+1)$

1

The matrix is

$a \ -1 \ 0$

$0 \ 1 \ 1$

$0 \ 0 \ 1$



Piotr Kałol April 22, 2015 at 10:30 PM

And can we solve $f(n) = a*n*f(n-1)+b*n*f(n-2)$? I tried treating n like a constant but it just doesn't work. And I couldn't transform it to any type mentioned by Zobayer.



Zobayer Hasan April 23, 2015 at 10:08 AM

@Piotr Kałol, I do not think it can be solved using matrix exponentiation methods, or at least not that I know of.



Piotr Kałol April 23, 2015 at 6:40 PM

Yeah, I figured that if it could be, you'd make more than 7 types. Anyway, thanks for the article. Very useful!

Yash June 22, 2015 at 4:29 PM

Matrix exponentiation is applicable only for solving linear recurrences (and recurrences convertible to linear ones).



Zobayer Hasan June 23, 2015 at 10:33 AM

Yash you are right.

[Reply](#)

Milind Mathur September 9, 2012 at 11:44 AM

Superb explanation. Very clear. Thanx a ton

[Reply](#)



techtechie September 27, 2012 at 2:58 AM

good job.....well written

[Reply](#)



Aman Verma October 28, 2012 at 2:52 AM

suberb... post.. and so simple to understand.. thanx...

[Reply](#)



Apoorv Upreti November 14, 2012 at 1:42 AM

Nicely written. There aren't many good tutorials on this topic.

[Reply](#)

Anonymous December 11, 2012 at 1:37 PM

Great post!!

[Reply](#)



Mohammad Samiul Islam February 11, 2013 at 8:05 PM

Great post. I managed to solve the first 2 :)

[Reply](#)

▼ [Replies](#)



Zobayer Hasan February 14, 2013 at 5:26 AM

Good luck with the rest too :) Thanks for visiting :)

[Reply](#)

Anonymous February 26, 2013 at 12:02 AM

Thanks a lot bro. I wanted to learn about this and at the right time found this blog.

[Reply](#)

▼ [Replies](#)



Zobayer Hasan February 28, 2013 at 8:55 PM

Thanks for visiting, I'm glad that it helped :)

[Reply](#)



Salman Estyak March 1, 2013 at 12:10 AM

```
while(1)
printf("ধ্বংসবাদ, ভাইয়া :)");
```

[Reply](#)

Anonymous March 3, 2013 at 9:45 PM

Hi , Thanks for writing this beautiful note :)

Can you please explain, how can we build the M for the following type of function? Thanks in advance.

```
g(n)=g(n-1)+g(n-2)+f(n-3)
where
f(n)=f(n-1)+f(n-2)+g(n-3)
```

[Reply](#)

▼ [Replies](#)



Zobayer Hasan March 3, 2013 at 9:59 PM

you can do it using a 6x6 matrix.
all you need to do is derive this matrix

$f(n+1)$

$f(n)$

$f(n-1)$

$g(n+1)$

$g(n)$

$g(n-1)$

and what you know is this matrix

$f(n)$

$f(n-1)$

$f(n-2)$

$g(n)$

$g(n-1)$

$g(n-2)$

so the matrix will be

1 1 0 0 0 1

1 0 0 0 0 0

0 1 0 0 0 0

0 0 1 1 1 0

0 0 0 1 0 0

0 0 0 0 1 0

[Reply](#)

Anonymous March 4, 2013 at 2:32 PM

I'm having trouble understanding the Formation of this matrix ...

$f(n+1)$

$f(n)$

$f(n-1)$

$g(n+1)$

$g(n)$

$g(n-1)$

I'm writing my reasoning , please correct it

I need to find , $f(n+1)$...

$f(n+1)=f(n)+f(n-1)+g(n-2)$

Here , $f(n+1)$ function's dependencies are $f(n)$, $f(n-1)$ and $g(n-2)$... now , I need to find out the dependencies of $g(n-2)$ to , get the value of $g(n-2)$, $g(n-2)=g(n-3)+g(n-4)+f(n-5)$ // $g()$ again is related to $f()$, I'm finding this confusing//

then , $f(n+1)$ total dependencies are $f(n), f(n-1), g(n-3), g(n-4), f(n-5)$ so , A becomes

a matrix with all the up stated values , and B will be a matrix with same functions but '1' added to the arguments . Please , Explain the fault in the reasoning a give a bit detail on what is really going on there .

Thanks .

Reply

▼ Replies



Zobayer Hasan March 4, 2013 at 8:32 PM

Yes, there are some mistakes in your reasoning, I mean, you are correct about the dependencies, but the fact is, if you already know what $g(n-2)$ is, you wont need to expand it further. In matrix exponentiation algorithm, with $M \cdot A = B$ form, the 1 column matrix you showed is B here, and A is what you already have. No need to follow recursions any more. Just try to test if recurrences hold. As I have written above, just multiply it and see for yourself you u can get the right side or not.

first row is 1 1 0 0 0 1, and the 1 column A matrix is

$f(n)$

$f(n-1)$

$f(n-2)$

$g(n)$

$g(n-1)$

$g(n-2)$

so, multiplying, we get the first row of B which is

$f(n)*1 + f(n-1)*1 + f(n-2)*0 + g(n)*0 + g(n-1)*0 + g(n-2)*1$

$\Rightarrow f(n) + f(n-1) + g(n-2)$

$\Rightarrow f(n+1)$.

similarly, second row is 1 0 0 0 0 0, this is $f(n)$

from third row, 0 1 0 0 0 0, we get $f(n-1)$

from fourth row 0 0 1 1 1 0, we get $f(n-2) + g(n) + g(n-1) \Rightarrow g(n+1)$

from fifth row 0 0 0 1 0 0, we get $g(n)$

from sixth row, 0 0 0 0 1 0, we get $g(n-1)$

so, multiplying the matrix M, we get

$f(n+1)$

$f(n)$

$f(n-1)$

$g(n+1)$

$g(n)$

$g(n-1)$

Now, imagine, if we already know

$f(n)$

$f(n-1)$

$f(n-2)$

$g(n)$

$g(n-1)$

$g(n-2)$

we can then find M^n to get $f(n+1)$ and $g(n+1)$ at the same time, don't follow typical recurrence reasoning here. I think the later part of the post describes how this is done actually. I don't know how can I make it more clear.



Zobayer Hasan March 4, 2013 at 8:35 PM

Correction, I wanted to write this in the last:

Now, imagine, if we already know

$f(2)$

$f(1)$

$f(0)$

$g(2)$

$g(1)$

$g(0)$.

then we can find M^n to get $f(2+n)$, $g(2+n)$ at the same time. the other values are just carried with the process.

Anonymous March 4, 2013 at 11:46 PM

Thanks , I think I got it finally .

:) I was taking the whole thing as the formal recursion and was trying to trace back and ...from $f()$ I was trying to trace back through $g()$ again $f()$...all that made me confused ...

sorry but I will nag you one more time :) I'm writing another recurrence function , please say whether it's okay or not :)

Let ,

$f(n)=f(n-1)+g(n-2)+h(n-3)$

$g(n)=g(n-1)+h(n-2)+f(n-3)$

$h(n)=h(n-1)+f(n-2)+g(n-3)$

then ,

B will be

$|f(n+1)|$
 $|f(n)|$
 $|f(n-1)|$
 $|g(n+1)|$
 $|g(n)|$
 $|g(n-1)|$
 $|h(n+1)|$
 $|h(n)|$
 $|h(n-1)|$

A will be :

$|f(n)|$
 $|f(n-1)|$
 $|f(n-2)|$
 $|g(n)|$
 $|g(n-1)|$
 $|g(n-2)|$
 $|h(n)|$
 $|h(n-1)|$
 $|h(n-2)|$

So , M will be

$|100010001|$
 $|100000000|$
 $|010000000|$
 $|001100010|$
 $|000100000|$
 $|000010000|$
 $|010001100|$
 $|000000100|$
 $|00000010|$

:) Thanks



Zobayer Hasan  March 5, 2013 at 12:26 AM

Yeah, its correct!

[Reply](#)

Anonymous March 30, 2013 at 11:52 AM

it really helped me to get thrgh the concept so easily..thank u...

[Reply](#)



Ravi Shankar October 15, 2013 at 2:45 AM

Thanks Sir for this post.

[Reply](#)

Anonymous October 29, 2013 at 11:52 PM

দাৰ্জলিং আইডিআ...

[Reply](#)

Anonymous December 23, 2013 at 11:54 AM

please give some idea about the prerequisite

[Reply](#)



Swagnik Dutta January 14, 2014 at 5:30 PM

That was an awesome article. However i failed to understand a certain part in type 7. I was hoping if you could explain how to arrive at this equation $g(n+1) = 2g(n) + 2g(n-1) + f(n+1)$. from the main equation : $g(n) = 2g(n-1) + 2g(n-2) + f(n)$, where, $f(n) = 2f(n-1) + 2f(n-2)$.

I could form the equations for all the other cases/types ,but I don't know how to work with equations with more than one recurrence. It would be very helpful if you could explain a bit briefly.

[Reply](#)

▼ Replies



Zobayer Hasan  January 14, 2014 at 10:20 PM

Hello there, about the recursion, isn't that putting $n+1$ instead of n ?

[Reply](#)



Swagnik Dutta January 15, 2014 at 1:29 AM

Sorry my previous question was rather silly. I don't understand why there are 4 rows in matrix A and B. I will try to explain what i have understood. Please rectify me if I'm wrong anywhere.

Given:

$$g(n) = 2g(n-1) + 2g(n-2) + f(n) \rightarrow (1)$$

$$f(n) = 2f(n-1) + 2f(n-2) \rightarrow (2)$$

replacing n by (n+1) in equation (1).

$$g(n+1) = 2g(n) + 2g(n-1) + f(n+1)$$

RHS of the equation is the present set of states. so they should be in matrix A.

hence matrix A should be:

| g(n) |

| g(n-1) |

| f(n+1) |

and matrix B should be containing the next set of states:

| g(n+1) |

| g(n) |

| f(n+2) |

All the types from 1 to 6 were solved this way, but here in case 7 the fourth row was introduced.

Can you tell me the reason behind that or why its necessary?

[Reply](#)

▼ Replies



Swagnik Dutta January 17, 2014 at 11:46 PM

I was hoping if you could clear my doubt mentioned above. :)



Zobayer Hasan January 18, 2014 at 12:19 AM

Sorry for late response, got carried away by the chores of daily life.

Anyway, its nothing different from the other types. Look closely, to know $f(n+1)$, you need to know both $f(n)$ and $f(n-2)$. So, if you only use three rows, how are you going to get the value of $f(n-2)$? So, you will need two $g()$ s and two $f()$ s in the matrix.

Hope it is clear now. The thing is, you need to have all the dependencies on the LHS matrix.



Zobayer Hasan January 18, 2014 at 12:20 AM

Oops! sorry about the type, its not $f(n-2)$ its $f(n-1)$. I think you got that as well,

[Reply](#)



TUSHAR PODDAR January 18, 2014 at 5:22 PM

What about $f(x,y)=f(x,y-1)+f(x-1,y)$?

It can be done using DP, but if x and y are as large as 10^6 then one has to look around! Matrix Expo possible?

[Reply](#)



TUSHAR PODDAR January 18, 2014 at 5:23 PM

What about $f(x,y)=f(x,y-1)+f(x-1,y)$?

It can be done using DP, but if x and y are as large as 10^6 then one has to look around! Matrix Expo possible?

[Reply](#)

▼ Replies



Zobayer Hasan January 24, 2014 at 3:37 AM

No, I don't think it's possible, or at least not in my knowledge.

[Reply](#)



TUSHAR PODDAR January 23, 2014 at 2:40 PM

Any hope with Matrix Expo? Another property of the function $f(x,y)$ was $f(x,y)=f(y,x)$

[Reply](#)

▼ Replies



Zobayer Hasan January 24, 2014 at 3:36 AM

I don't think that can be done using matrix exponentiation. If you can express one variable in terms of the other one, then some thoughts can be made, but I have no clue how to handle function on more than one variable recurrences using matrix exponentiation.

[Reply](#)



Anchit Jain April 2, 2014 at 8:45 PM

Please explain Type 7 . It's given that $g(n) = 2g(n-1) + 2g(n-2) + f(n)$ and $f(n) = 2f(n-1) + 2f(n-2)$.

So $g(n+1) = 2g(n) + 2g(n-1) + f(n+1)$ and $f(n+1) = 2f(n) + 2f(n-1)$

so shouldn't matrix A be $[g(n), g(n-1), f(n), f(n-1)]$.

Please explain how did you got $|g(n)|$

$|g(n-1)|$

$|f(n+1)|$

$|f(n)|$

Someone also has asked this doubt but I could not understand your explanation for that. Please explain asap.

[Reply](#)



\$ATYA October 14, 2014 at 11:11 PM

Nice Article!

Can u please tell how to make approach for 11651?

Thanks

[Reply](#)



Shivam October 28, 2014 at 2:39 AM

Nice article...can u please share ur email address!!

[Reply](#)



Mahesh Dhoni November 10, 2014 at 9:46 AM

zobayer why are we considering $f(n)$ & $f(n-1)$ in matrix exponentiation after all we want to find $f(n+1)$

[Reply](#)

▼ Replies



Zobayer Hasan November 10, 2014 at 11:33 AM

do you mean why we are keeping them on right hand side? they are generated as partial result, but we are not worried about them, as long as we find $f(n+1)$.

but on the left side matrix, we need them both because $f(n+1)$ depends on both.

[Reply](#)

Anonymous January 25, 2015 at 12:27 PM

$T(n) = \max(a + T(n-b), c + T(n-d))$.

how can we solve this with matrix exponentiation ?

[Reply](#)

▼ Replies



Zobayer Hasan April 23, 2015 at 10:14 AM

Seems more like dynamic programming to me. I have never tried anything of this sort, lets see if someone else can help you.

[Reply](#)



Ishu Garg June 13, 2015 at 10:31 PM

Man read this topic for the first time and you nailed it. Keep it up. And thanks for writing such a brilliant post. Are there more of such useful kinds... looking forward for reading them too. Thanks

[Reply](#)

Anonymous September 23, 2015 at 7:38 PM

Hi Zobayer Hasan,

I am new to competitive programming but good at DS and Algo concepts. Can u please tell me where to start. Please share your experiences how have u mastered all tricks. How many years does it take. How much time do we need to spend coding.

[Reply](#)



vsn harish rayasam September 23, 2015 at 7:39 PM

Can you give one tutorial like this for dynamic programming. Greedy and for graph algorithms. It would be of great help

[Reply](#)

▼ Replies



Zobayer Hasan September 24, 2015 at 10:16 PM

Well, I could find some for you searching google but then I wouldn't know if the material is good or not. To be honest, learning something deeply cannot be achieved by reading random tutorials, casually learnt knowledge does not persist much long in our brain, but the knowledge you gain when you get stuck with a problem goes a long way. My suggestion would be, do not waste time following tutorials, follow problems, and if you cant solve them, then find those solutions, you'll learn lot faster, in my opinion. :)

That being said, there are good tutorials at <http://topcoder.com/tc>

Reply



Karan Kapoor January 23, 2016 at 3:30 PM

Hey Can you Please explain the matrix for odd even case. I could not figure it out :(

[Reply](#)

Anonymous January 26, 2016 at 12:35 PM

hi very nice article - what about $f(2n) = f(n) + f(n-1) + n$ for $n > 1$ and $f(2n+1) = f(n-1) + f(n) + 1$ for $n \geq 1$?? any thoughts appreciated

[Reply](#)

Anonymous January 26, 2016 at 12:44 PM

oh i forgot to mention $f(0) = 1$, $f(1) = 1$, and $f(2) = 2$ - thank you

[Reply](#)



Aditi Kaushik March 12, 2016 at 9:27 PM

hey ! Zobayer, thanks a ton man ! a

[Reply](#)

Anonymous March 25, 2016 at 3:00 AM

Just wanted to say that this was a really helpful, clearly-explained post! Great work!

[Reply](#)



Unknown June 21, 2016 at 10:26 PM

Great Notes ,thankx.

[Reply](#)



nitish kumar July 7, 2016 at 2:45 AM

It was a great Help . Thanks Man :)

[Reply](#)

Md. Shahedul Islam Shahed December 10, 2016 at 11:55 AM

ভাইয়া,

UVa 11651 : Krypton Number System পুরোনোটাতে ম্যাক্সিমাম লেংগথ $1e9$ না হয়ে আরো কম হলে dp দিয়ে সলভ করতে পারতাম, কিন্তু ওটাতে লিনিয়ার রিকার্সেস কিভাবে হচ্ছে সেটা ধরতে পারতেছি না বলে matrix exponentiation এপ্লাই করতে পারছি না... কইডলি যদি হেল্প করতেন... :)

[Reply](#)



deepak gautam December 24, 2016 at 3:31 AM

$f(n) = f(n-1) + f(n-2)$ if (n is odd)

$f(n) = f(n-1) - f(n-2)$ if (n is even)

how can we solve this with matrix exponentiation ?

[Reply](#)



Ovishek Paul January 16, 2017 at 7:25 PM

ম্যাট্রিক্স এক্সপোনেনশিয়েশন বোঝার জন্য এটা একটা সবথেকে BEST reference.. thank you!

[Reply](#)



Ovishek Paul January 16, 2017 at 7:26 PM

Step 6 টা কিয়া যদি একই কিস্তারিত লিখতেন তাহলে ভালো হইত (আলাদা আলাদা ব্যাপারটা বুঝিনাই) ...

[Reply](#)



Vikas Dubey June 27, 2017 at 12:02 AM

<http://codeforces.com/problemset/problem/821/E> can you explain formation of matrix for this question

[Reply](#)

Enter your comment...



Comment as: Google Account ▾

Publish

Preview

Newer Post

Home

Older Post

Subscribe to: [Post Comments \(Atom\)](#)

CATAGORIES

[academic study \(23\)](#) [access modifiers \(1\)](#) [ajax \(1\)](#) [algorithm \(53\)](#) [analysis \(7\)](#) [apache \(1\)](#) [backtrack \(1\)](#) [bash \(1\)](#) [beginner \(17\)](#) [bfs \(2\)](#) [bigint \(1\)](#) [binary indexed tree \(2\)](#) [binary tree \(1\)](#) [bit \(1\)](#) [bitbucket \(1\)](#) [bitvector \(1\)](#) [blogger \(5\)](#) [bpm \(3\)](#) [brainfuck \(1\)](#) [brute force \(1\)](#) [bst \(1\)](#) [c \(5\)](#) [c++ \(41\)](#) [changes \(1\)](#) [character device driver \(1\)](#) [chat \(3\)](#) [client \(3\)](#) [combinatorics \(2\)](#) [command prompt \(1\)](#) [common \(1\)](#) [comparator \(1\)](#) [compression \(1\)](#) [computational geometry \(3\)](#) [confusion \(1\)](#) [connected component \(1\)](#) [console \(1\)](#) [constructible polygon \(1\)](#) [contest \(12\)](#) [crc \(1\)](#) [cse \(5\)](#) [css \(1\)](#) [customize \(1\)](#) [data mining \(2\)](#) [data structure \(16\)](#) [database \(3\)](#) [DCEPC206 \(1\)](#) [decoding \(1\)](#) [design \(1\)](#) [device driver \(1\)](#) [dfs \(1\)](#) [disjoint set \(1\)](#) [divide and conquer \(3\)](#) [dp \(4\)](#) [driver \(1\)](#) [dual boot \(1\)](#) [dynamic programming \(10\)](#) [encoding \(1\)](#) [encryption \(1\)](#) [error \(2\)](#) [esoteric language \(2\)](#) [euler circuit \(3\)](#) [euler path \(2\)](#) [euler phi \(1\)](#) [example \(1\)](#) [execution \(1\)](#) [expression evaluation \(1\)](#) [extended euclid \(1\)](#) [facebook \(3\)](#) [factorization \(2\)](#) [fibonacci \(1\)](#) [fix time \(1\)](#) [function \(1\)](#) [funny \(15\)](#) [gcd \(2\)](#) [geometry \(6\)](#) [git \(3\)](#) [github \(2\)](#) [gns3 \(2\)](#) [graph \(9\)](#) [GUANGGUN \(1\)](#) [hacker cup \(2\)](#) [hashing \(1\)](#) [hiding window \(1\)](#) [hints \(16\)](#) [holi \(1\)](#) [hopcroft karp \(1\)](#) [huffman \(1\)](#) [incorrect clock \(1\)](#) [inner class \(1\)](#) [instance \(1\)](#) [jar \(1\)](#) [java \(8\)](#) [javascript \(2\)](#) [jdbc \(1\)](#) [jquery \(1\)](#) [kernel programming \(2\)](#) [lab \(5\)](#) [lazy propagation \(1\)](#) [like \(1\)](#) [linear algebra \(1\)](#) [linux \(6\)](#) [ls \(1\)](#) [makefile \(1\)](#) [math \(21\)](#) [matrix \(3\)](#) [matrix algebra \(2\)](#) [matrix exponentiation \(2\)](#) [matrix multiplication \(2\)](#) [maxflow \(2\)](#) [maximum bipartite matching \(3\)](#) [maximum flow \(2\)](#) [merge sort \(3\)](#) [mistake \(1\)](#) [modular arithmetic \(2\)](#) [module compiling \(2\)](#) [multichat \(3\)](#) [mysql \(1\)](#) [networking \(2\)](#) [number system \(1\)](#) [number theory \(8\)](#) [online judge \(4\)](#) [operating system \(1\)](#) [os \(1\)](#) [other \(8\)](#) [parallel programming \(3\)](#) [pattern \(1\)](#) [phi \(1\)](#) [poj \(1\)](#) [polconst \(1\)](#) [pollard rho \(1\)](#) [practice \(2\)](#) [primes \(5\)](#) [primit \(1\)](#) [priority queue \(1\)](#) [problem \(17\)](#) [problem classifier \(4\)](#) [problem solving \(57\)](#) [problems solving \(1\)](#) [programming \(70\)](#) [pruning \(1\)](#) [pthreads \(4\)](#) [puzzle \(1\)](#) [python \(1\)](#) [qualification round \(1\)](#) [queen \(1\)](#) [queue \(1\)](#) [range maximum query \(1\)](#) [recursion \(6\)](#) [reflection \(1\)](#) [repository \(4\)](#) [rip \(1\)](#) [rmq \(1\)](#) [sample \(1\)](#) [segment tree \(2\)](#) [server \(3\)](#) [shell \(1\)](#) [shell script \(1\)](#) [sieve \(4\)](#) [simulation \(3\)](#) [socket \(3\)](#) [solution \(3\)](#) [spacing \(1\)](#) [sphere online judge \(28\)](#) [spoj \(29\)](#) [static routing \(1\)](#) [syntax highlighting \(1\)](#) [system programming \(4\)](#) [table tag \(1\)](#) [tc \(1\)](#) [template \(4\)](#) [thread \(3\)](#) [time mismatch \(1\)](#) [time setting \(1\)](#) [topcoder \(2\)](#) [topology \(2\)](#) [training \(6\)](#) [tree \(3\)](#) [tutorial \(10\)](#) [ubuntu \(1\)](#) [usaco \(2\)](#) [uva \(5\)](#) [uva online judge \(5\)](#) [vector \(1\)](#) [version control \(1\)](#) [web server \(1\)](#) [windows \(3\)](#)

I am only one, but still I am one.
I cannot do everything, but still I can do something.
And because I cannot do everything I will not refuse to do the something that I can do.
{Helen Keller}

Picture Window theme. Theme images by Ollustrator. Powered by Blogger.