⟵ Notes

Exponentiation (Integer/Matrix) (Useful in Competitive Programming)

7

Competitive Programming   Competitive coding   Math

2 LIVE EVENTS

Lets say in a programming contest you need to find a^b % MOD .

One of the naive methods is to run a loop from 1 to b , keep multiplying and mod

```
prod = 1;
for(i=1;i<=b;i++)
{
prod*=a;
prod%=MOD;
}
```

This is done in order O(b) as it requires multiplying a b times if b is as big as 10^6 or more it may give a **Time Limit Excedded** error

Lets look at it in an another way

Example You want to calculate 2^32 you can do divide and conquer

```
2^1     =   2
2^2     = (2^1)^2     = 4
2^4     = (2^2)^2      = 16
2^8     = (2^4)^2      = 256
2^16   = (2^8)^2      = 65536
2^32   = (2^16)^2    = 4294967296
```

Insted of 32 steps we found this in only 5 multiplication steps thus we can reduce a O(b) problem to an O(log(b)) problem

Below is the code for Modular Integer exponentiaion in C++ for caluclating (a^p)%mod

```
#include<iostream>
using namespace std;
long long int exp(long long int a, long long int p ,long long int
{
long long int result = 1;
```

```
 long long int (p==0)
    return 1;
 long long int (p==1)
    return a;
while(p)
{
    if(p&1)
        result *= a;
    result%=mod;
    p >>=1;
    a*=a;
    a%=mod;
}
return result;
}
int main()
{
    long long int a,p,m;
    m=1;
    cout<<"Enter a , p and mod : ";
    cin>>a>>p>>m;
    cout<<exp(a,p,m)<<'\n';
    return 0;
}
```

This can be extended to matrix also, insted of integer multiplication call a matrix multiplication (useful in Graph problems) Below is a C++ code for matrix multiplication

```
#include<iostream>
#include<algorithm>
#include<cstdio>
#include<cstring>
#include<vector>
#include<map>
#define mod 1000000007
#define ull unsigned long long int
#define fl(i,n) for(i=0;i<n;i++)
#define pn printf('\n')
#define ps printf(' ')
using namespace std;
unsigned long long int** mmul(unsigned long long int** m,unsigned
{
    unsigned long long int i,j,k,**result;
    result = new unsigned long long int* [N];
    fl(i,N)
    result[i]=new unsigned long long int[N];
    fl(i,N)
    fl(j,N)
    {
        result[i][j]=0;
        fl(k,N)
        result[i][j]+=m[i][k]*n[k][j];
    }
        return result;
}
unsigned long long int** mpow(unsigned long long int** matrix,unsi
{
    unsigned long long int **m,i,j;
    m = new unsigned long long int* [n];
        fl(i,n)
            m[i]=new unsigned long long int[n];
        fl(i,n)
            fl(j,n)
            {
                if(i==j)
                    m[i][j]=1;
                else
                    m[i][j]=0;
            }
    if(p==0)
        return m;
    else if(p==1)
        return matrix;
    while(p)
    {
        if(p&1)
            m = mmul(m,matrix,n);
        p>>=1;
            matrix = mmul(matrix,matrix,n);
    }
    return m;
}
int main()
```

```
{
    unsigned long long int n,i,j,p;
    unsigned long long int** matrix;
    cout<<"Enter N  : ";
    cin>>n;
            matrix = new unsigned long long int* [n];
    fl(i,n)
    matrix[i]=new unsigned long long int[n];
    cout<<"Enter Elements :\n";
    fl(i,n)
    fl(j,n)
    cin>>matrix[i][j];
    cout<<"Enter Power : ";
    cin>>p;
    matrix=mpow(matrix,p,n);
    cout<<'\n';
    fl(i,n)
    {
        fl(j,n)
        cout<<matrix[i][j]<<' ';
        cout<<'\n';
    }
    return 0;
}
```

Like 0 Tweet 1 G+1 0

AUTHOR

Arun Prasad
Student at IIIT Bangalore
Bangalore
4 notes

Write Note

My Notes

Drafts

TRENDING NOTES

Strings And String Functions
written by Vinay Singh

Segment Tree and Lazy Propagation
written by Akash Sharma

Number Theory - II
written by Tanmay Chaudhari

Matrix exponentiation
written by Mike Koltsov

Graph Theory - Part II
written by Pawel Kacprzak

more ...

ABOUT US

Blog

Engineering Blog

Updates & Releases

Team

Careers

In the Press

HACKEREARTH

API

Chrome Extension

CodeTable

HackerEarth Academy

Developer Profile

Resume

Campus Ambassadors

Get Me Hired

Privacy

Terms of Service

DEVELOPERS

AMA

Code Monk

Judge Environment

Solution Guide

Problem Setter Guide

Practice Problems

HackerEarth Challenges

College Challenges

RECRUIT

Developer Sourcing

Lateral Hiring

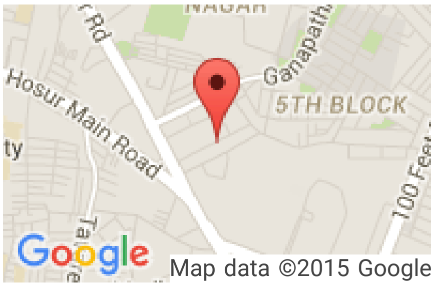
Campus Hiring

FAQs

Customers

Annual Report

REACH US



IIIrd Floor, Salarpuria Business Center,
4th B Cross Road, 5th A Block,
Koramangala Industrial Layout,

Bangalore, Karnataka 560095, India.

✉ contact@hackerearth.com

☎ +91-80-4155-4695

☎ +1-650-461-4192