

EOlymp 11289 Labeled Graphs

Nama : Timothy Hosia Budianto

NRP : 50525211098

Permasalahan dan pendekatan :

Labeled graphs

Let the number of vertices in a graph be n . Compute the number of labeled graphs with n vertices (labeled means that the vertices are marked with the numbers from 1 to n). The edges of the graphs are considered undirected, and loops and multiple edges are forbidden.

Input

The number of vertices n ($1 \leq n \leq 10^9$) in the graph.

Output


Print the number of labeled graphs with n vertices. Print the answer modulo $10^9 + 7$.

 Time limit 1 second

 Memory limit 128 MiB

Input example #1 


2

Output example #1 

2

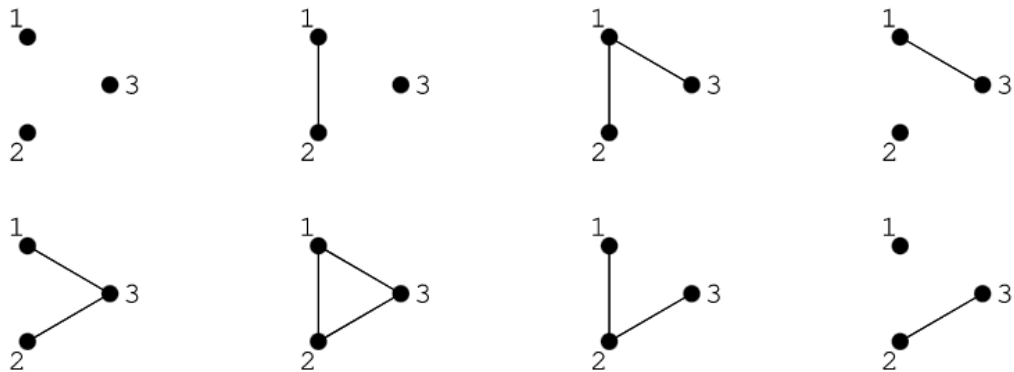
Input example #2 

3

Output example #2 

8

Soal ini menanyakan ada berapa set of graph edges yang bisa dibuat, given jumlah titik yang ada. Maksud soal dapat lebih jelas terlihat dari ilustrasi berikut dengan test case 2 (3 titik) akan menghasilkan 8 cara.



Sumber [Labeled Graph -- from Wolfram MathWorld](#)

Absraksi :

Dengan konsep diatas kita akan menghitung hasil – hasil lainnya yaitu dari 1 titik hingga 6 titik sebagai berikut.

input	output	
1	1	2^0
2	2	2^1
3	8	2^3
4	64	2^6
5	1024	2^{10}
6	32768	2^{15}

Dilihat bahwa hasilnya adalah hasil dari perpangkatan bilangan 2 dengan sebuah barisan bilangan. Disini kita dapat menerapkan konsep **divide and conquer** untuk mencari rumus dari barisan bilangan tersebut.

Terlihat bahwa rumus dari baris bilangan tersebut **sesuai dengan rumus bilangan triangular** yaitu rumusnya adalah

$$\frac{n(n-1)}{2}$$

Jika diteruskan barisan dari triangular number adalah

List Of Triangular Numbers

0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153, 171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496, 528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035, 1081, 1128, 1176, 1225, 1275, 1326, 1378, 1431, and so on.

Sumber dari [Triangular Numbers Sequence - List and Formula \(byjus.com\)](https://byjus.com/maths/triangular-numbers/)

Jadi dapat disimpulkan bahwa rumus yang dihasilkan adalah

$$a(n) = 2^{(n*(n-1)/2)}.$$

Implementasi :

Dengan pseudocode :

Main

Input n

If(n=1) return 1

Else

Ans = $2^{(n(n-1)/2)}$ modulo 1000000007

Output Ans

Source Code :

```
#include <cstdio>
using namespace std;
#define gc getchar
#define ULL unsigned long long
typedef long long ll;

ULL read(){
    ULL value = 0; bool ne=0;
    char c = gc();
    while(c==' ' or c=='\n') c = gc();
    if(c=='-'){ne = 1; c = gc();}
    while(c>='0' and c<='9'){
        value = (value<<3)+(value<<1)+c-'0'; c = gc();}
    if(ne) value*=-1;
    return value;
}

ll modex(ll b, ll e, ll m){
    if (b == 0 || e == 0) return 0;
    ll r =1;
```

```

while (e>0){
    if ((e&1)==1) r = (r*b)%m;
    e>>=1;
    b = (b*b)%m;
}
return r;
}

int labeledgraphs(ULL n)
{
    long long ans;
    if(n==1)return 1;
    else ans = modex(2,(n*(n-1)/2),1000000007);
    return ans;
}

int main()
{
    ULL n = read();
    printf("%lld", labeledgraphs(n));
    return 0;
}

```

Penjelasan code :

Line 1-5 adalah deklarasi yang akan digunakan

Line 7-16 fungsi read untuk bypass handler

Line 18-27 adalah fungsi untuk memangkatkan dan memodulo yang akan di inputkan dengan penjelasan parameter 2 pangkat $(n*(n-1)/2)$ di modulo 1000000007.

Line 37-43 adalah fungsi main dimana akan meng inputkan nilai ke variable n, kemudian akan di passing ke fungsi labeledgraphs untuk diproses.

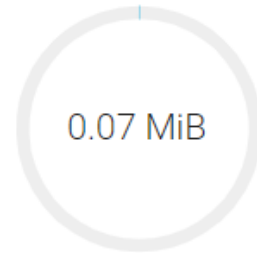
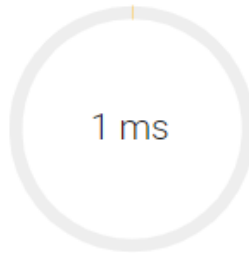
Hasil e-olymp :

Problem[Labeled graphs](#)**Submitted**

1 hour ago

Programming Language

C++ 17 (gnu 10.2)

Author[TimothyHosia_5025211098](#)

Your submission was graded A+, which means it passed all tests and used LESS resources than 99% of the submissions on the website.

Test #	Status	Score	Duration	CPU	Memory
✓ Test suite #1	Accepted	150 / 150	1 ms	1 ms	76 KiB
✓ Test #1	Accepted	10 / 10	1 ms	1 ms	76 KiB
✓ Test #2	Accepted	10 / 10	1 ms	1 ms	68 KiB
✓ Test #3	Accepted	10 / 10	1 ms	1 ms	68 KiB
✓ Test #4	Accepted	10 / 10	1 ms	1 ms	72 KiB
✓ Test #5	Accepted	10 / 10	1 ms	1 ms	68 KiB