Recursión

Recursión es Recursión

PCFIM

17 de enero de 2023

¿Qué es una recursión?

¿Qué es una recursión?

Recursión

> Una recursión es una técnica matemática utilizada en programación. Se dice que una función es recursiva cuando se llama a sí misma.

> Una recursión puede definirse como una:
>
> ——recursión

Una recursión simple

```cpp
#include <bits/stdc++.h>
using namespace std;

int Fibo(int n){
    if(n <= 1) return n;
    return Fibo(n-1) + Fibo(n-2);
}

int main(){
    cout << Fibo(60);
    return 0;
}
```

Problemas:

- Muy lento: $\mathcal{O}(\varphi^n)$.
- Recursión muy profunda.
- Overflow (?)

Mejorando la recursión

```cpp
#include <bits/stdc++.h>
using namespace std;
#define N 10000
int f[N];
int Fibo(int n){
    if(n <= 1) return n;
    if(f[n] != 0) return f[n];
    f[n] = Fibo(n-1) + Fibo(n-2);
    return f[n];
}

int main(){
    cout << Fibo(60);
    return 0;
}
```

Problemas:

- ~~Muy lento: $\mathcal{O}(\varphi^n)$.~~
- Recursión muy profunda.
- Overflow (?)

Problemas :D

Dado un entero N, imprimir el fractal de Sierpinski de profundidad N.

Sierpinski de profundidad 1:

Sierpinski de profundidad 2:

Fractal de Sierpinski

```cpp
void fractal(int n, int h, int l){
    if (n == 1){
        mat[h][l] = '/'; mat[h][l+1] = '\\';
        mat[h+1][l-1] = '/'; mat[h+1][l] = '_';
        mat[h+1][l+1] = '_'; mat[h+1][l+2] = '\\';
        return;
    }
    fractal(n-1, h, l);
    fractal(n-1, h + (1<<n-1), l - (1<<n-1));
    fractal(n-1, h + (1<<n-1), l + (1<<n-1));
}

int main(){
    memset(mat, ' ', sizeof mat);
    int n = 5;
    fractal(n, 1, 1<<n);
    for (int i = 1; i <= (1<<n); i++){
        for (int j = 1; j <= (1<<n) + i; j++) cout << mat[i][j];
        cout << '\n';
    }
    cout << '\n';
    return 0;
}
```

Thanos Sort

Thanos sort es un algoritmo de ordenación de supervillanos, que funciona de la siguiente manera: si el array no está ordenado, chasquea los dedos para eliminar la primera o la segunda mitad de los elementos y repite el proceso. Dado un array de entrada, ¿cuál es el tamaño del array ordenado más largo que puede obtener usando Thanos sort?

Se requiere Infinity Gauntlet.

Thanos Sort

```cpp
#include <bits/stdc++.h>
using namespace std;
int solve(int l, int r, vector <int> &arr){
    if(r - l == 0) return 0;
    bool found = true;
    for(int i=l; i<r; i++){
        if(arr[i] > arr[i+1]){
            found = false;
            break;
        }
    }
    if(found) return r-l;
    else return(max(solve(l, l+(r-l)/2, arr), solve(l+(r-l)/2+1, r, arr)));
}
int main(){
    int n;
    cin >> n;
    vector <int> arr(n);
    for(int i=0; i<n; i++) cin >> arr[i];
    cout << 1+solve(0, n-1, arr);
    return 0;
}
```

Un poquito de backtracking

Dado un tablero de ajedrez ordinario (8×8), de cuantas formas es posible colocar 8 reinas en el tablero, tal que ninguna esté atacando a otra.

N Reinas

```
void solve(vector <int> &table, int p){
    if(p == N-1){
        print(); return;
    }
    for(int j=0; j<N; j++){
        table[p+1] = j;
        bool cancel = false;
        for(int i=0; i<p+1; i++){
            if(table[i] == table[p+1]){
                cancel = true; break;
            }
            if(abs(table[p+1] - table[i]) == p+1-i){
                cancel = true; break;
            }
        }
        if(cancel) continue;
        solve(table, p+1);
    }
    table[p+1] = -1;
    return;
}
```