НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Algorithms & Programming
Programming Basics

C/C++/Kotlin programming
(p.3 – Functions - Recursion)

Yevhen Berkunskyi, NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

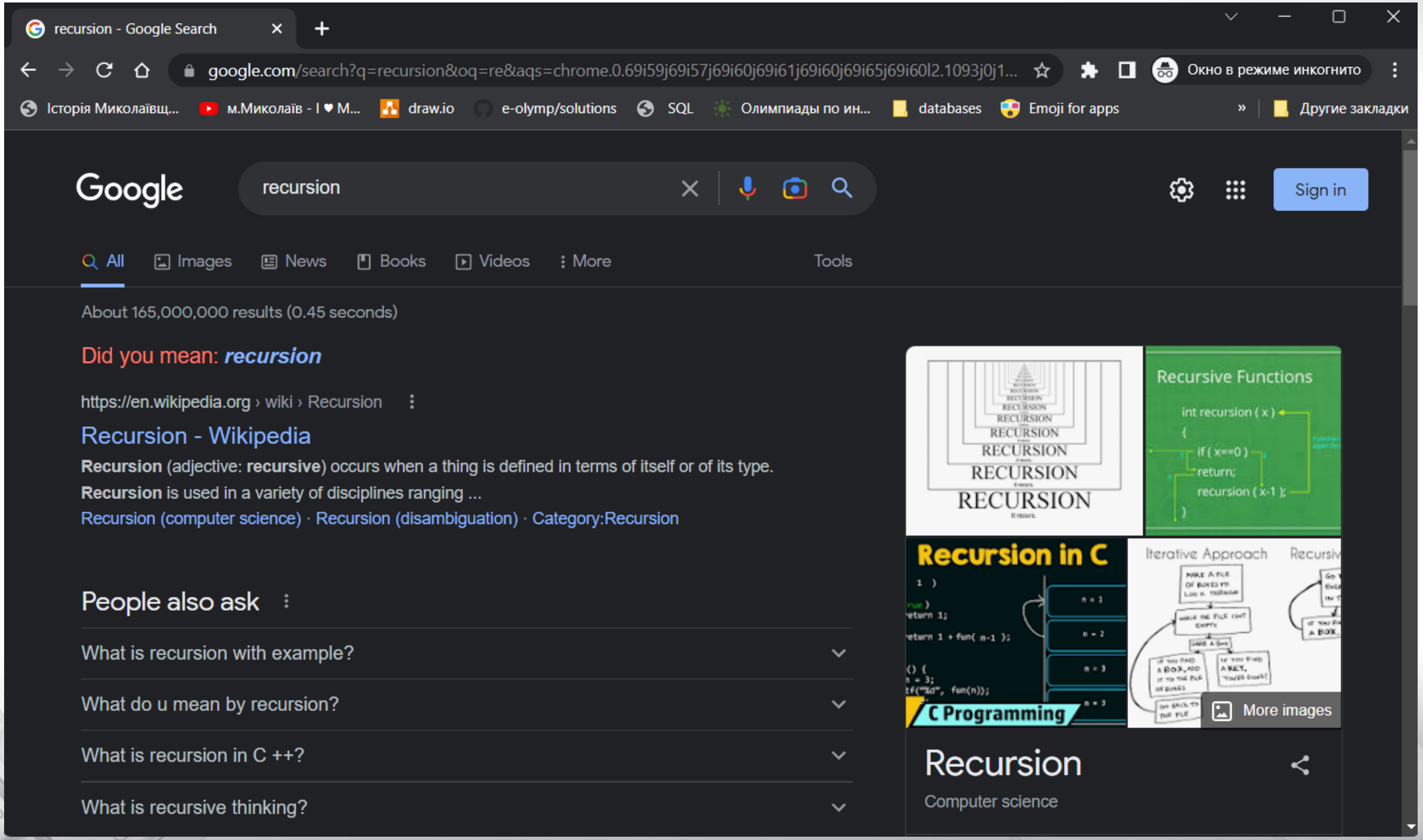
Recursion

What is recursion ?

Let’s use Google?..

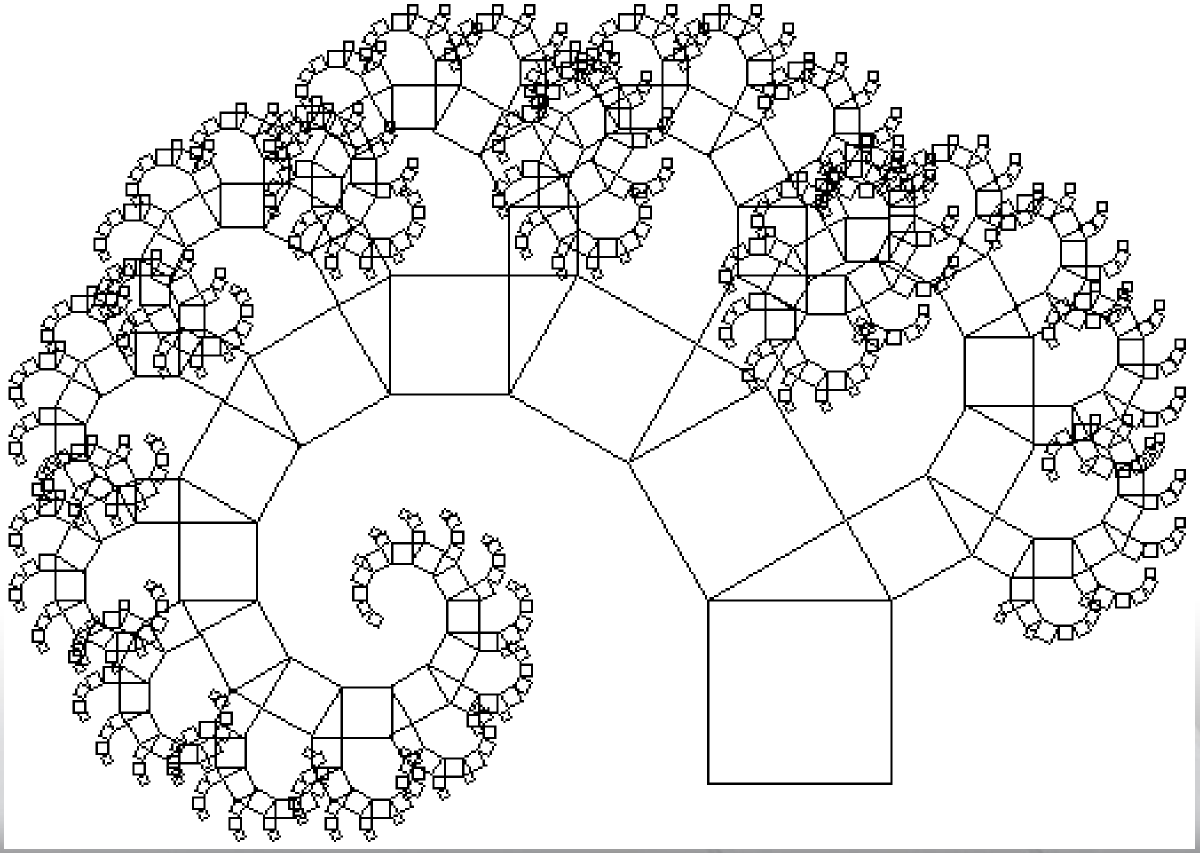
Recursion

Recursion is a ... ?



Recursion

Recursion is the definition of a part of a function through itself, that is, it is a function that calls itself, directly (in its body) or indirectly (through another function).



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Recursion

Typical recursive tasks are tasks:

- Caclulating n!
- Finding Fibonacci numbers.

Such tasks have already been solved by us, but only using cycles, that is, iteratively.

Generally speaking, everything that is solved iteratively can be solved recursively, that is, using a recursive function.

Recursion

Calculating factorial:

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Iterative (looping) process:

```
int f = 1;
for (int i = 1; i <= n; i++) {
    f *= i;
}
```

Recursion

Calculation of factorial

$$n! = \begin{cases} 1, & if\ n = 0\ or\ n = 1 \\ (n-1)! * n, & if\ \ n > 1 \end{cases}$$

Recursive process:

```
int f(int n) {
    if ( n == 0 || n == 1 ) return 1;
    else return n * f(n-1);
}
```

Calculation of factorial

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Tail Recursion

Tail recursion is defined as a recursive function in which the recursive call is the last statement that is executed by the function. So basically nothing is left to execute after the recursion call.

$$f(n,a) = \begin{cases} a, & n = 0 \\ f(n-1, n*a), & n > 0 \end{cases}$$

```
int Factorial(int n, int a = 1) {
   // return condition
   if (n==0)
      return a;
   // tail recursive call
   return Factorial(n - 1, n * a);
}
```

Recursion

Fibonacci numbers:

$$f_i = \begin{cases} 1, & if \;\; i = 0 \; or \; i = 1 \\ f_{i-1} + f_{i-2}, & if \; i > 1 \end{cases}$$

Fibonacci numbers

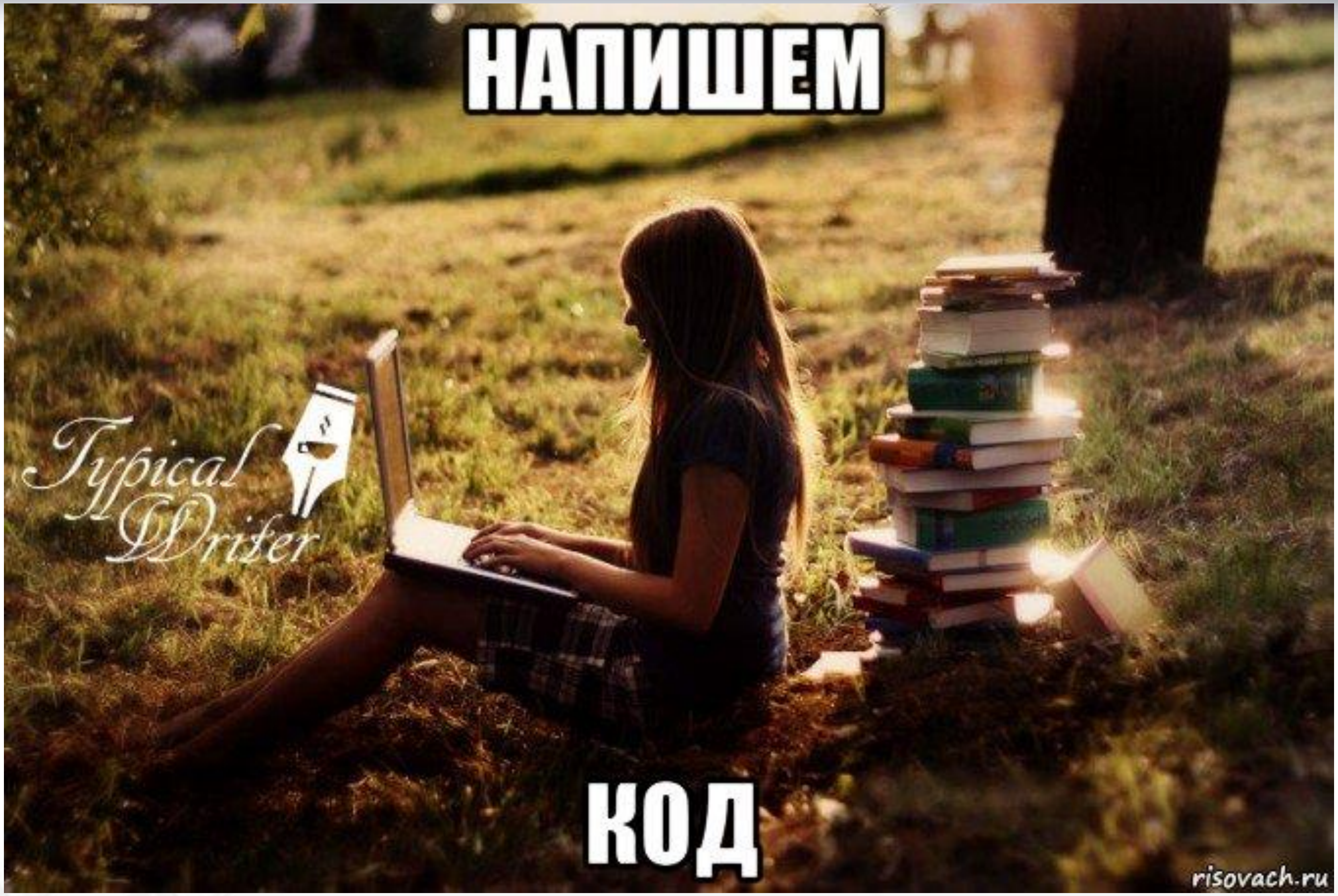
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Simple recursion:

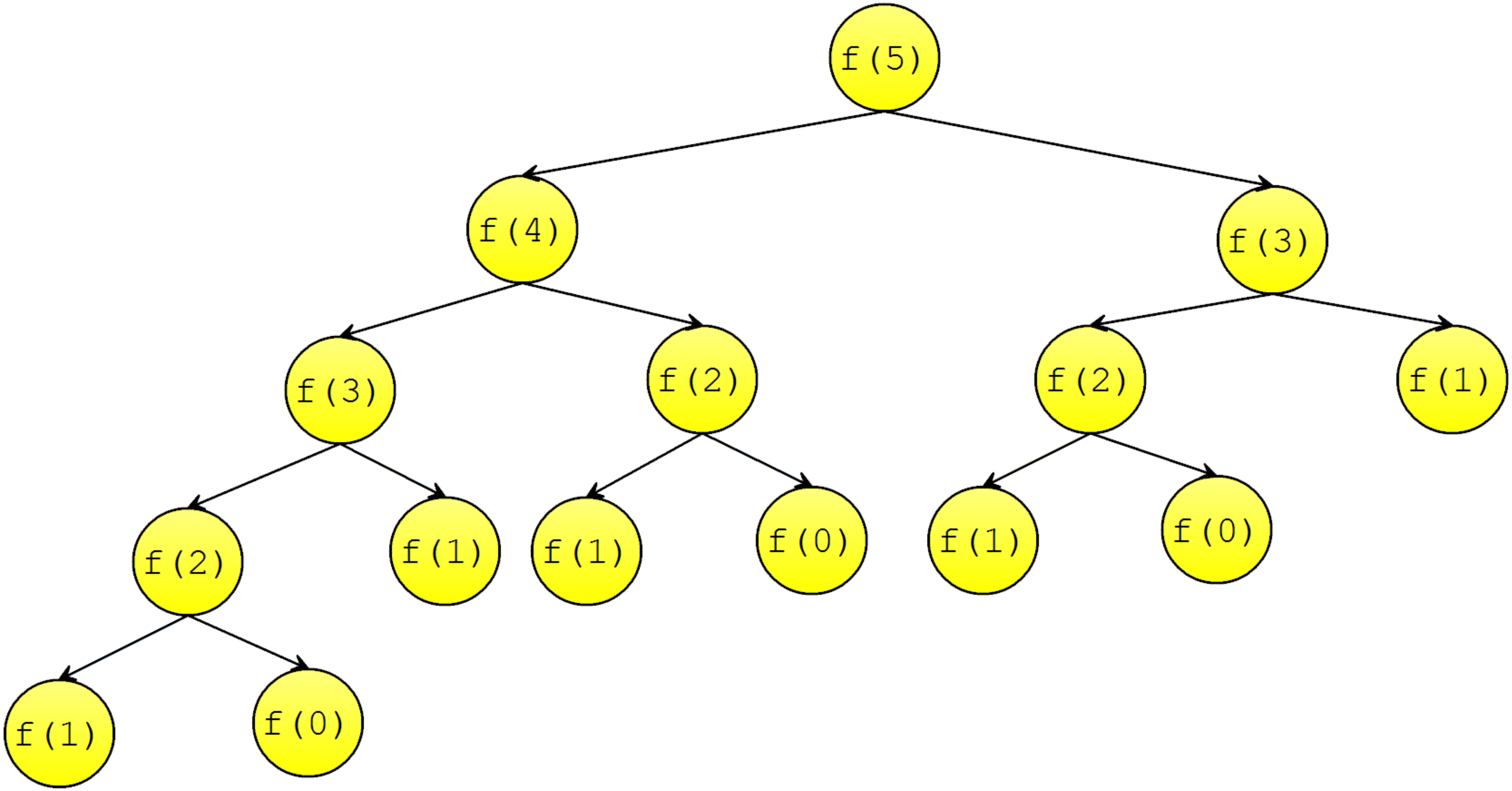
```
int fib(int i){
    if (i == 0 || i == 1) return 1;
    else return fib(i - 1) + fib(i - 2);
}
```

And what is wrong with it?

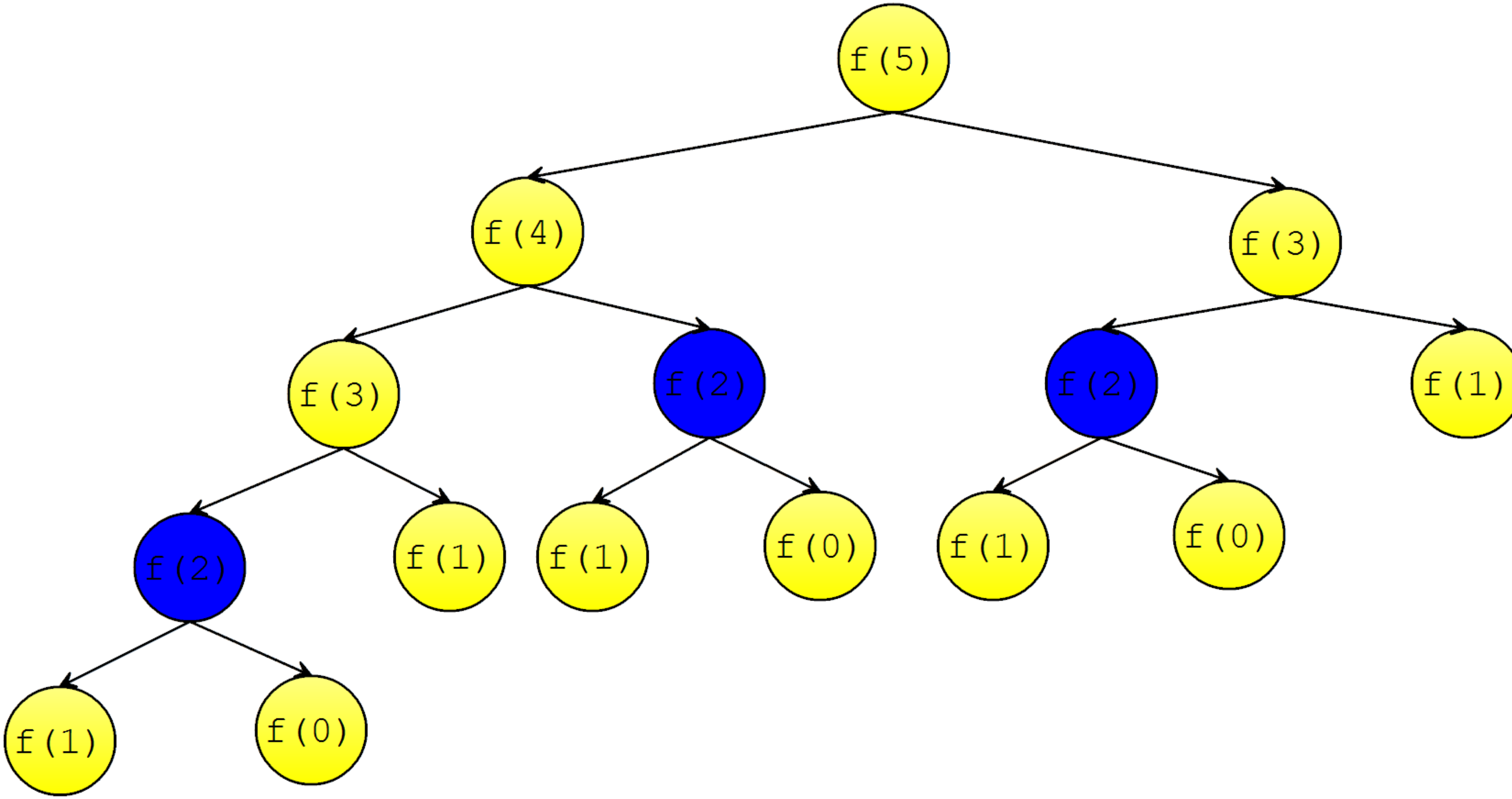
Demo



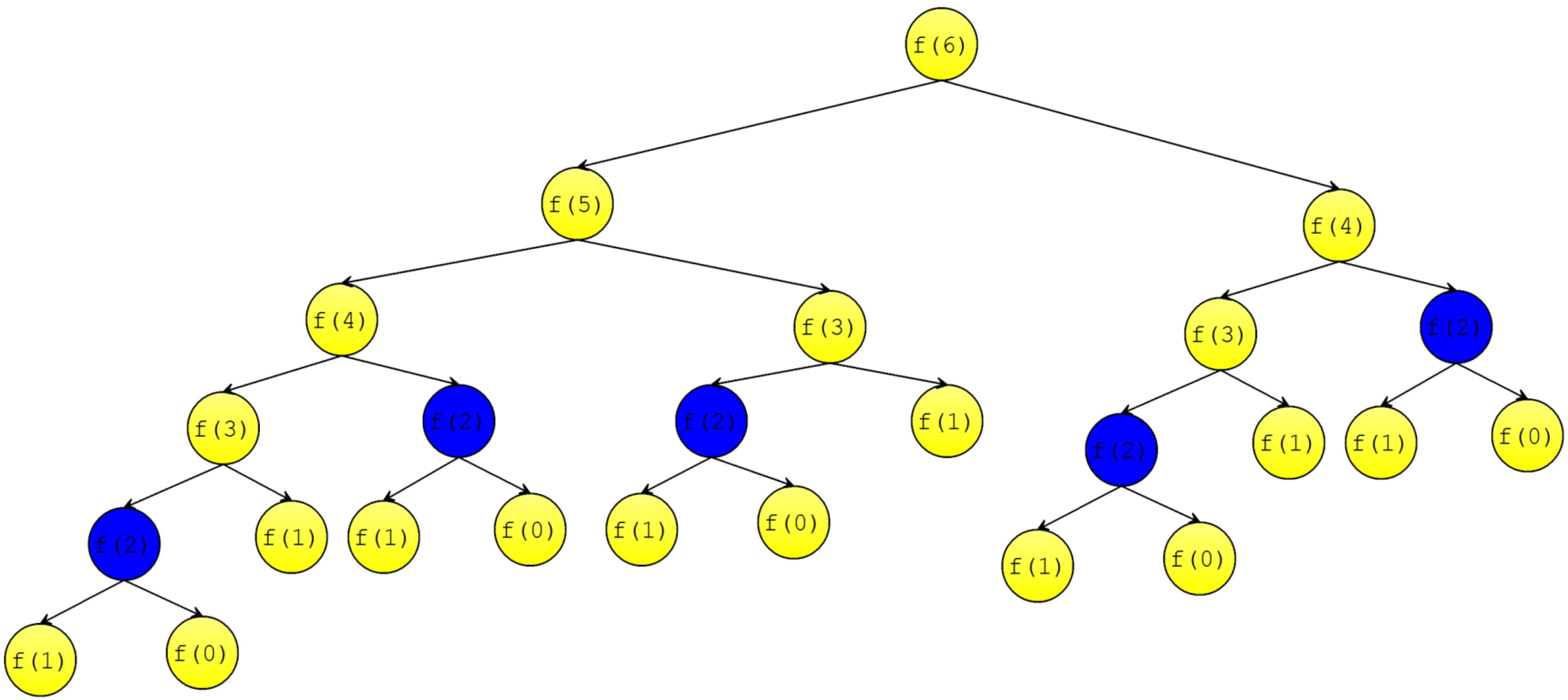
Recursion



Recursion



Recursion



We have a problem. How can we solve it?

Recursion

Tail recursion

```
int Fib(int n, int a = 0, int b = 1)
{
// return condition
   if (n == 1)
      return b;
   else
// tail recursive call
      return Fib(n-1, b, a+b);
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Recursion

And one more difficult problem.

And more interesting ☺

«Tower of Hanoi».

The objective of the puzzle is to move the entire stack to the last rod, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
- No disk may be placed on top of a disk that is smaller than it.

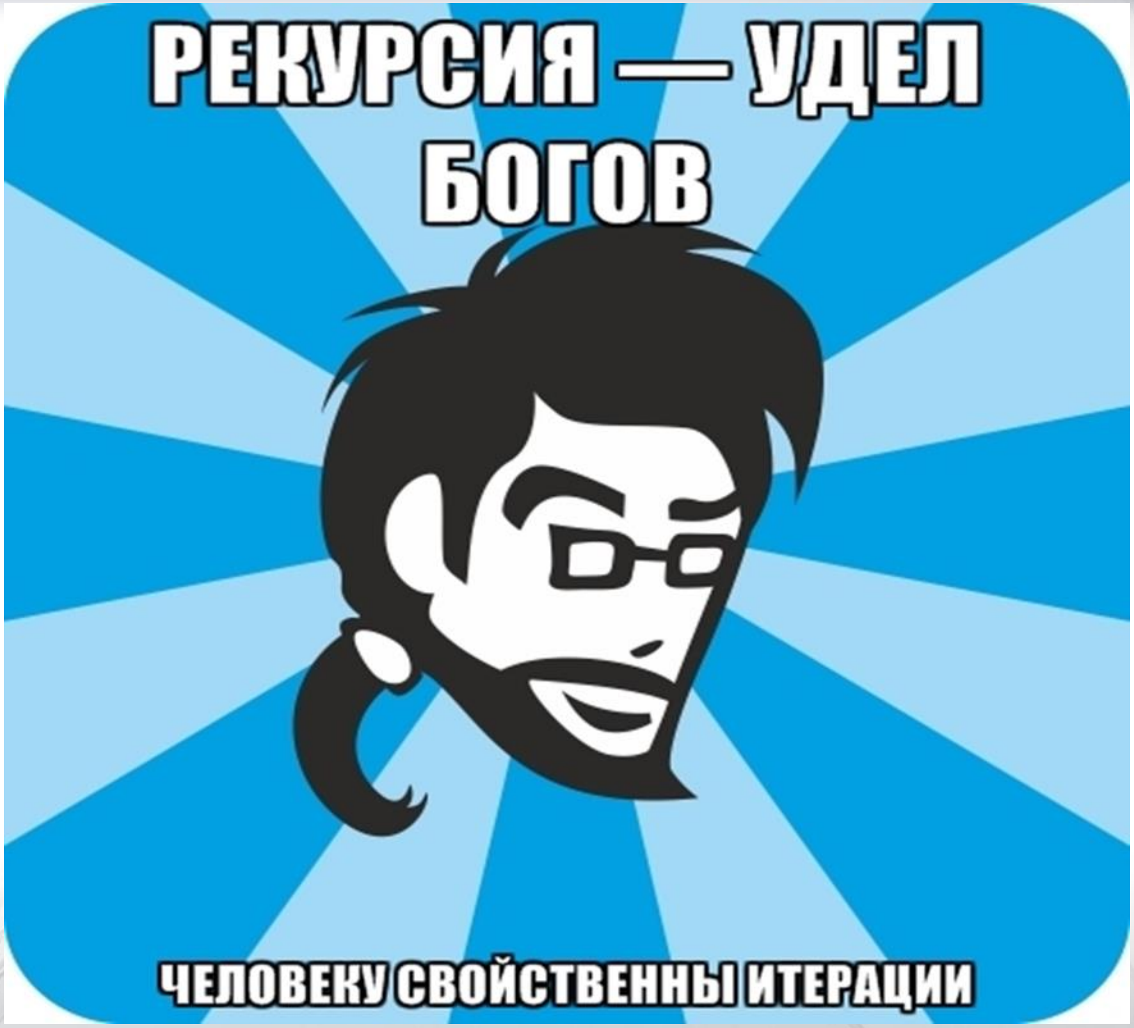
Recursion «Tower of Hanoi»

Tower of Hanoi

```
#include <iostream>

using namespace std;

void p(int n, int a, int b, int c) {
    if (n == 1) {
        cout << a << "->" << b << "\n";
    } else {
        p(n-1, a, c, b);
        p(1, a, b, c);
        p(n-1, c, b, a);
    }
}

int main() {
    int n;
    cin >> n;
    p(n, 1, 2, 3);
    return 0;
}
```

Recursion

Optimistic final ☺




QUESTIONS
& ANSWERS

НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Algorithms & Programming
Programming Basics

C/C++/Kotlin programming
(p.3 – Functions - Recursion)

Yevhen Berkunskyi, NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua