



9-11 классы

Программирование на C++

Презентация занятия

Рекурсия.

13 занятие



Минцифры
России



20.35
УНИВЕРСИТЕТ

Программирование
на C++

Теоретическая часть

Рекурсия.

13 занятие

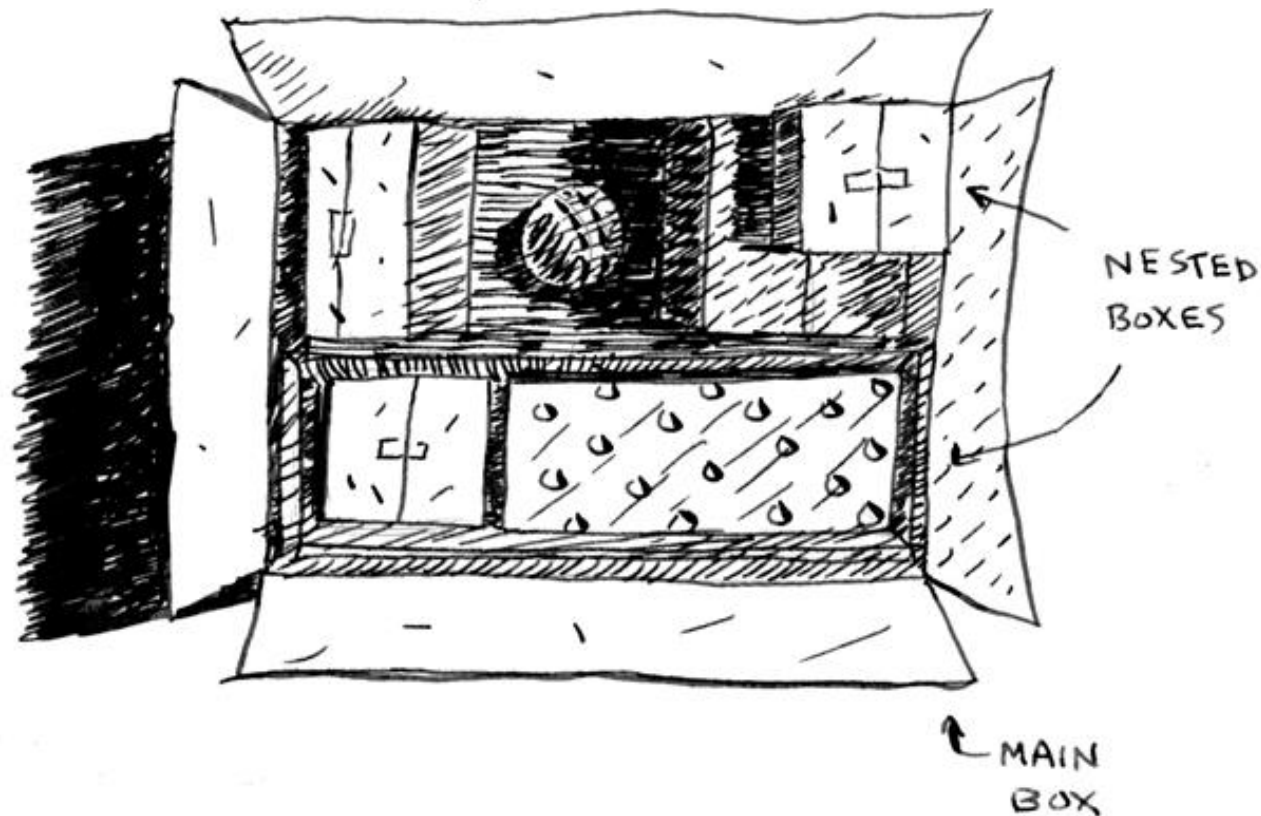


2020

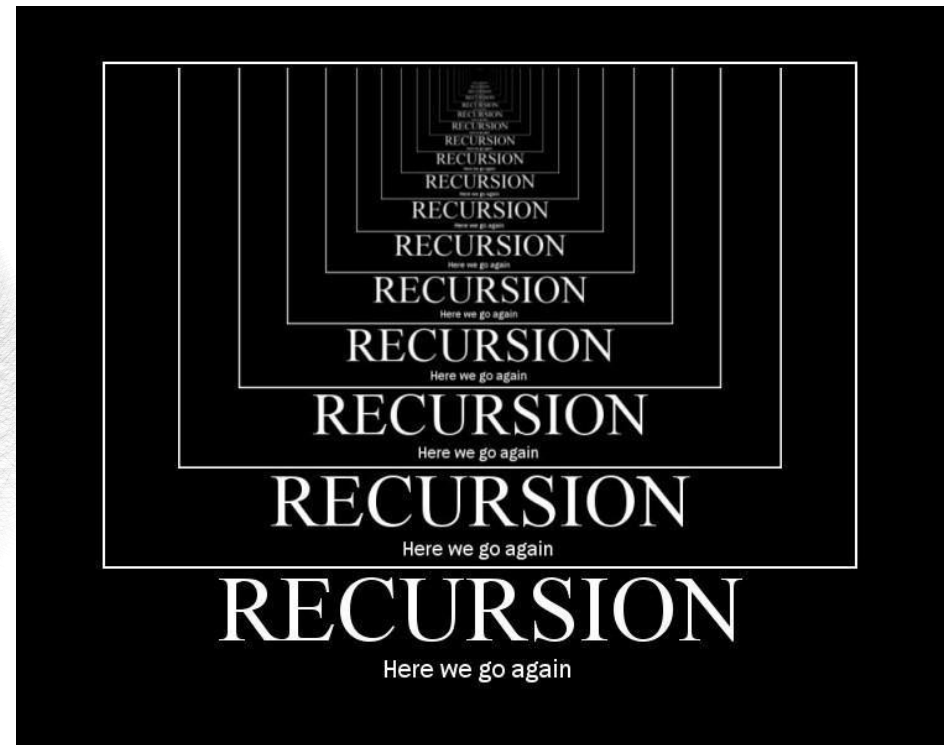
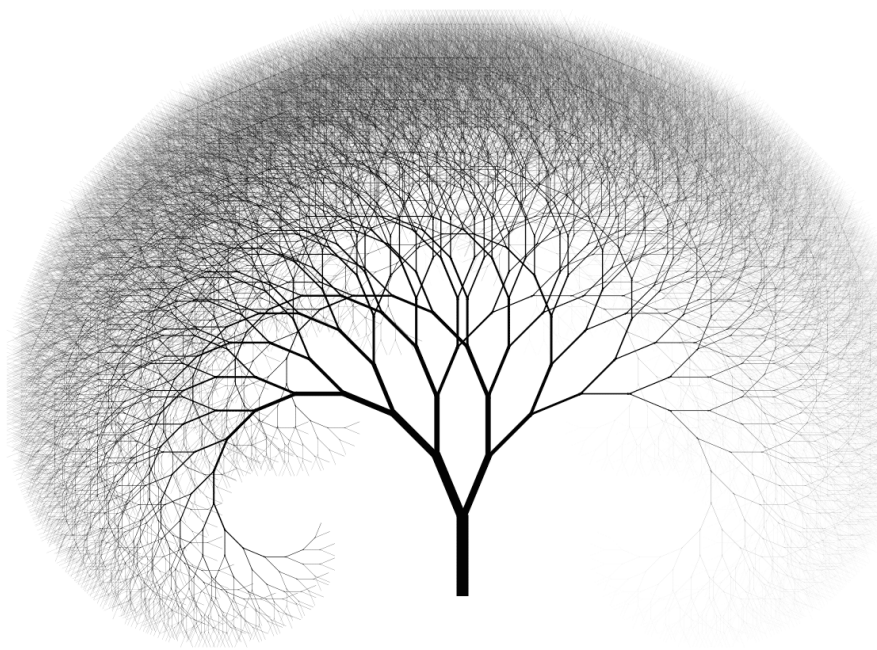


Тема: Рекурсия

“In order to understand recursion, one must first understand recursion.”



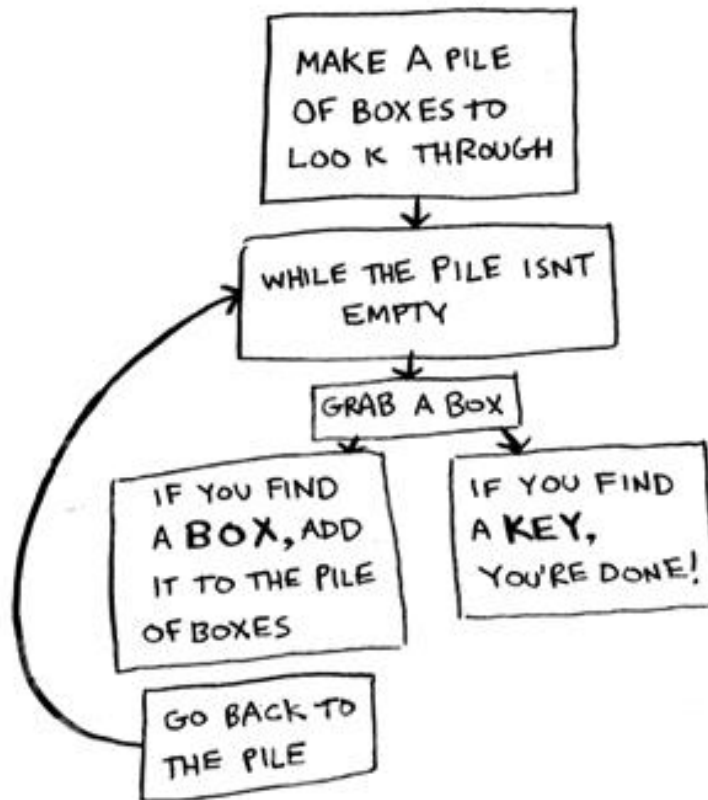
Тема: Рекурсия



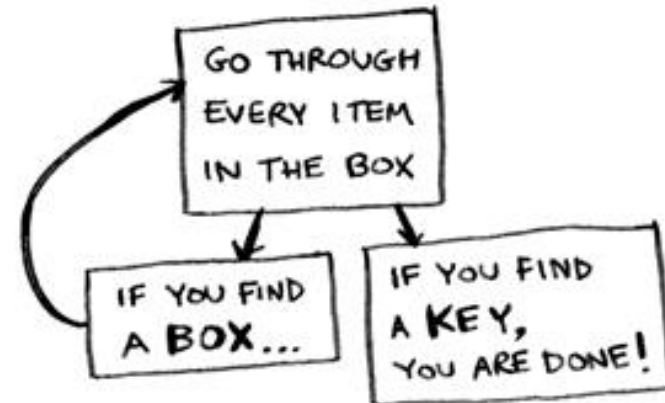


Тема: Рекурсия

Iterative Approach



Recursive Approach



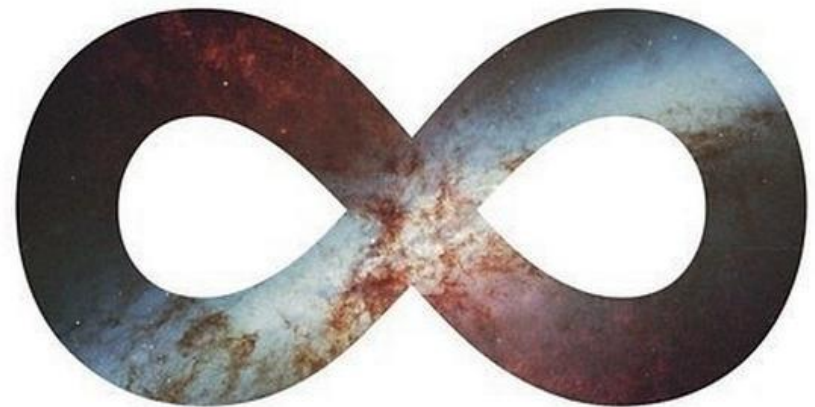
Тема: Рекурсия

Рекурсивная функция в языке C++ (и не только), это функция, которая вызывает саму себя.

Для избегания бесконечного вызова рекурсии, необходимо включить завершающее условие в функцию

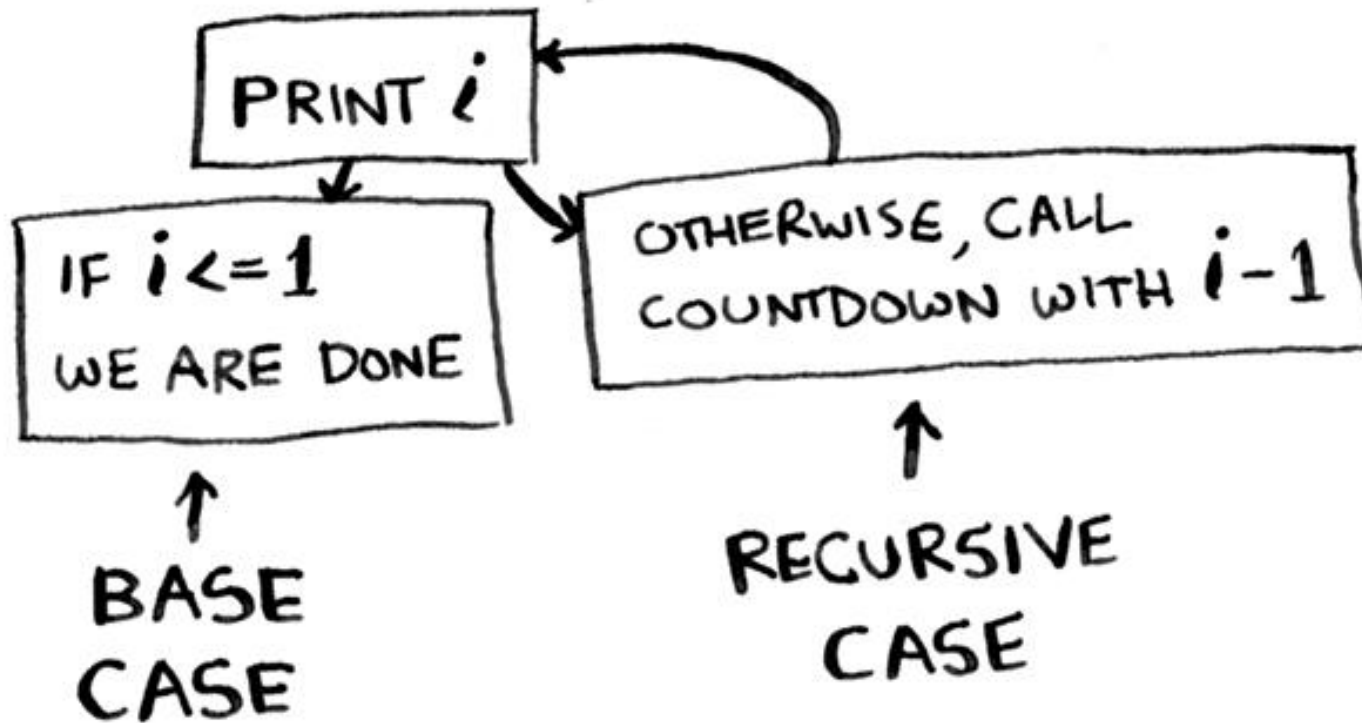
БЕСКОНЕЧНОСТЬ - НЕ ПРЕДЕЛ

```
1  void function()  
2  {  
3      |    function();  
4  }
```





Тема: Рекурсия



Тема: Рекурсия

Стек вызовов

Рекурсивные функции используют так называемый «Стек вызовов». Когда программа вызывает функцию, функция отправляется на верх стека вызовов. Это похоже на стопку книг, вы добавляете одну вещь за одни раз. Затем, когда вы готовы снять что-то обратно, вы всегда снимаете верхний элемент.



Тема: Рекурсия

Для демонстрации работы рекурсии рассмотрим, создадим программу для подсчёта **факториала**.

В математике термин факториал означает умножение всех положительных целых чисел, которые меньше, либо равны специальному неотрицательному (n)

$$1! = 1;$$

$$2! = 2 * 1;$$

$$3! = 3 * 2 * 1;$$

$$4! = 4 * 3 * 2 * 1;$$

$$0! = 1 \text{ (по определению)}$$

$n!$





Тема: Рекурсия

```
1  #include <iostream>
2  using namespace std;
3
4  int recursion(int num)
5  {
6      return (num ? num * recursion(num - 1): 1);
7      /*
8      if (!num) // num == 0
9      {
10         return (1);
11     }
12     else
13     {
14         return (num * recursion(num - 1))
15     }
16     */
17 }
18
19 int main()
20 {
21     cout << "Output:" << endl;
22     cout << recursion(3) << endl;
23     return (0);
24 }
```

Output:
6





Тема: Рекурсия

CODE CALL STACK

fact(3)

FACT
X 3

FIRST CALL TO fact.
X IS 3.

if x==1:

FACT
X 3

else:

FACT
X 3

return x * fact(x-1)

FACT
X 2
FACT
X 3

A RECURSIVE CALL!

NOW WE ARE IN
THE SECOND CALL
TO fact. X IS 2

if x==1:

FACT
X 2
FACT
X 3

THE TOPMOST FUNCTION
CALL IS THE CALL WE
ARE CURRENTLY IN

else:

FACT
X 2
FACT
X 3

NOTE: BOTH FUNCTION CALLS
HAVE A VARIABLE NAMED X
AND THE VALUE OF X
IS DIFFERENT IN BOTH

return x * fact(x-1)

FACT
X 1
FACT
X 2
FACT
X 3

YOU CAN'T ACCESS
THIS CALL'S X
FROM THIS CALL
AND VICE VERSA

if x==1:

FACT
X 1
FACT
X 2
FACT
X 3

WOW, WE MADE
THREE CALLS TO
fact, BUT WE
HAD NOT FINISHED
A SINGLE CALL UNTIL
NOW!

return 1

FACT
X 1
FACT
X 2
FACT
X 3

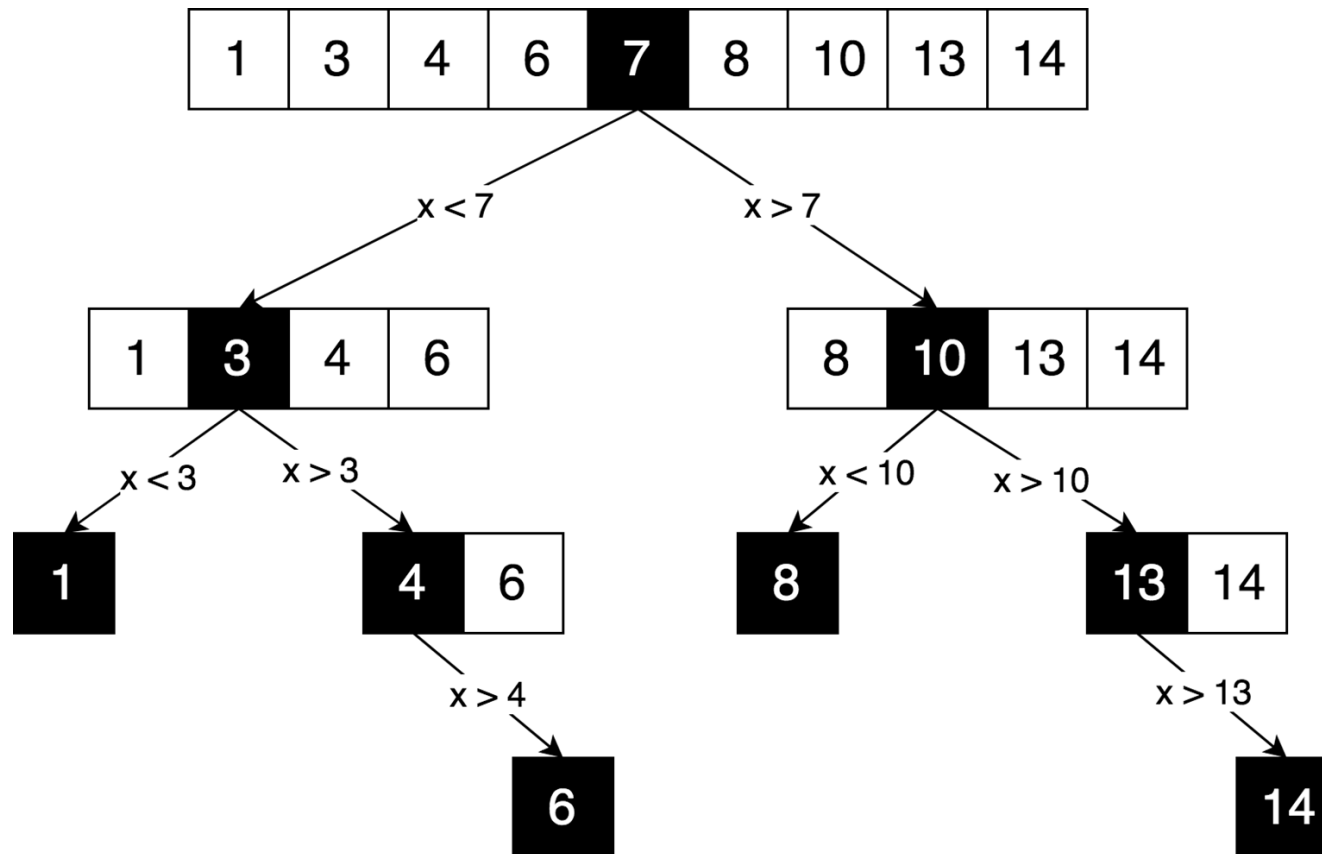
THIS IS THE FIRST BOX
TO GET POPPED OFF THE
STACK, WHICH MEANS
IT'S THE FIRST CALL WE
RETURN FROM

RETURNS 1



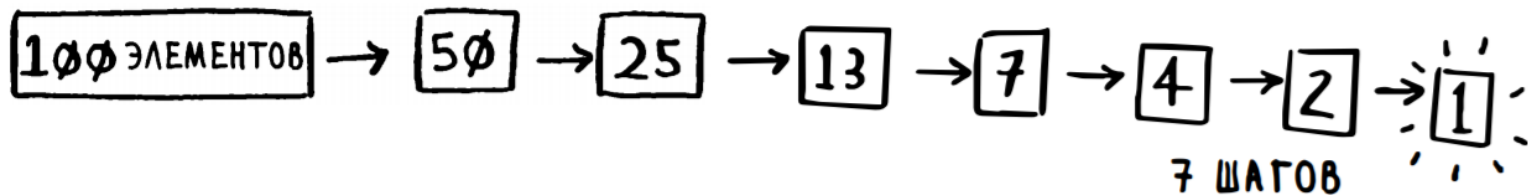
Тема: Рекурсия

Divide and Conquer

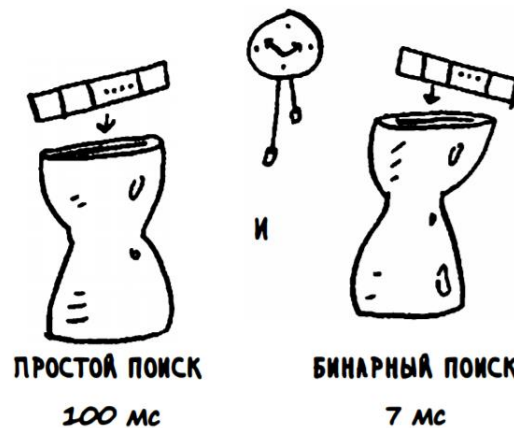




Тема: Рекурсия



При бинарном поиске каждый раз исключается половина чисел



Время выполнения простого и бинарного поиска для списка из 100 элементов





Тема: Рекурсия

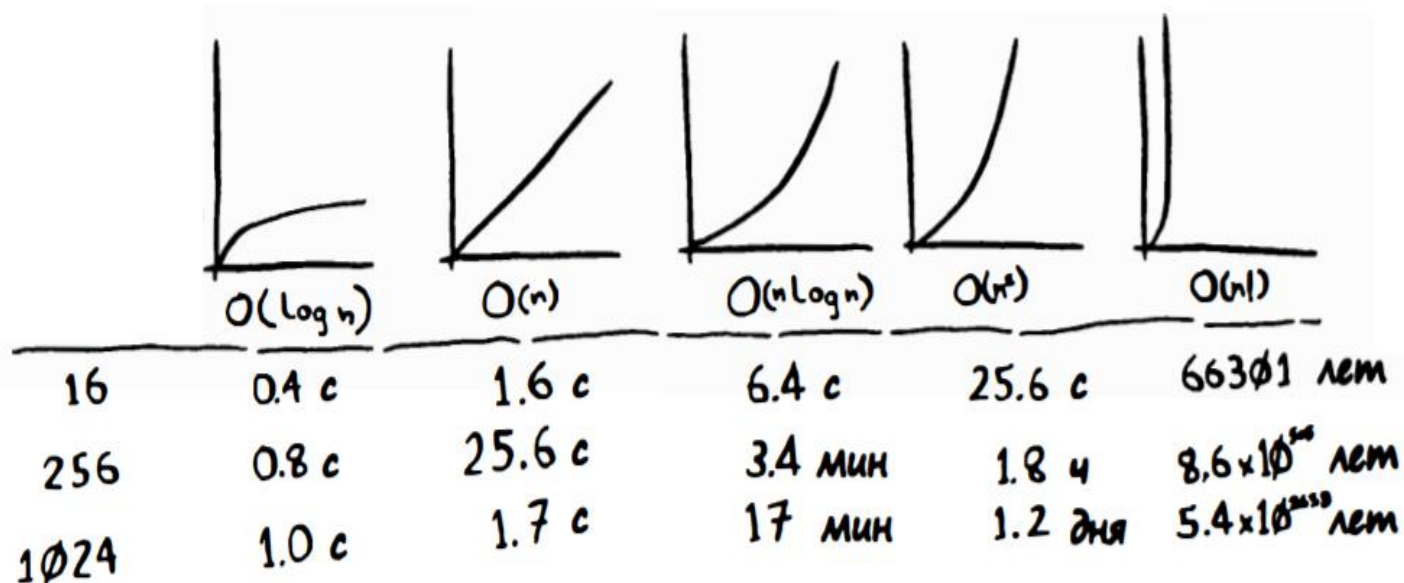
	ПРОСТОЙ ПОИСК	БИНАРНЫЙ ПОИСК
100 ЭЛЕМЕНТОВ	100 мс	7 мс
10 000 ЭЛЕМЕНТОВ	10 секунд	14 мс
1 000 000 ЭЛЕМЕНТОВ	11 дней	32 мс

Время выполнения растет с совершенно разной скоростью!





Тема: Рекурсия



Программирование
на C++

Практическая часть

Рекурсия.

13 занятие



2020

Тема: Рекурсия

Задание 1

Написать программу для подсчёта факториала числа
(итерационно, т е с помощью циклов)

$$1! = 1;$$

$$2! = 2 * 1;$$

$$3! = 3 * 2 * 1;$$

$$4! = 4 * 3 * 2 * 1;$$

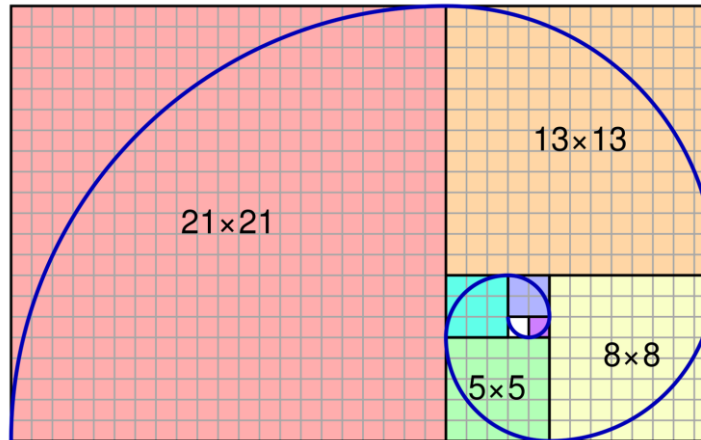
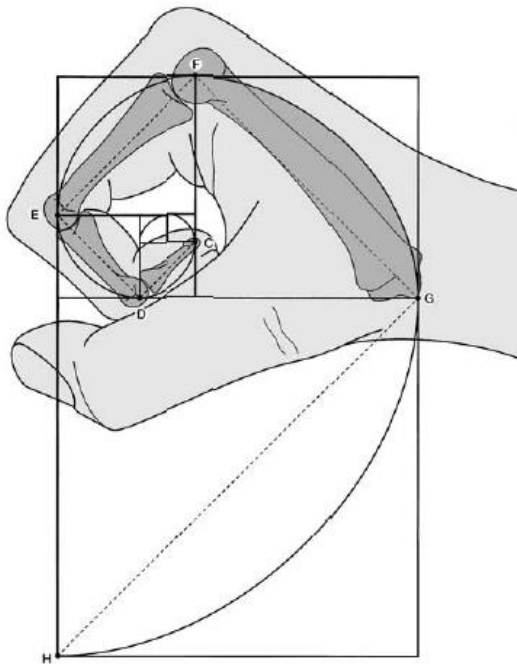
$$0! = 1 \text{ (по определению)}$$



Тема: Рекурсия

Задание 2

Вывести n-ое число
Фибоначчи



Сумма чисел Фибоначчи

$$\begin{aligned} 1 &= 1 \\ 1 + 1 &= 2 \\ 1 + 1 + 2 &= 4 \\ 1 + 1 + 2 + 3 &= 7 \\ 1 + 1 + 2 + 3 + 5 &= 12 \\ 1 + 1 + 2 + 3 + 5 + 8 &= 20 \\ 1 + 1 + 2 + 3 + 5 + 8 + 13 &= 33 \\ 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 &= 54 \\ 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 &= 88 \\ 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 + 55 &= 143 \\ 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 + 55 + 89 &= 232 \end{aligned}$$

