



Разработка приложений

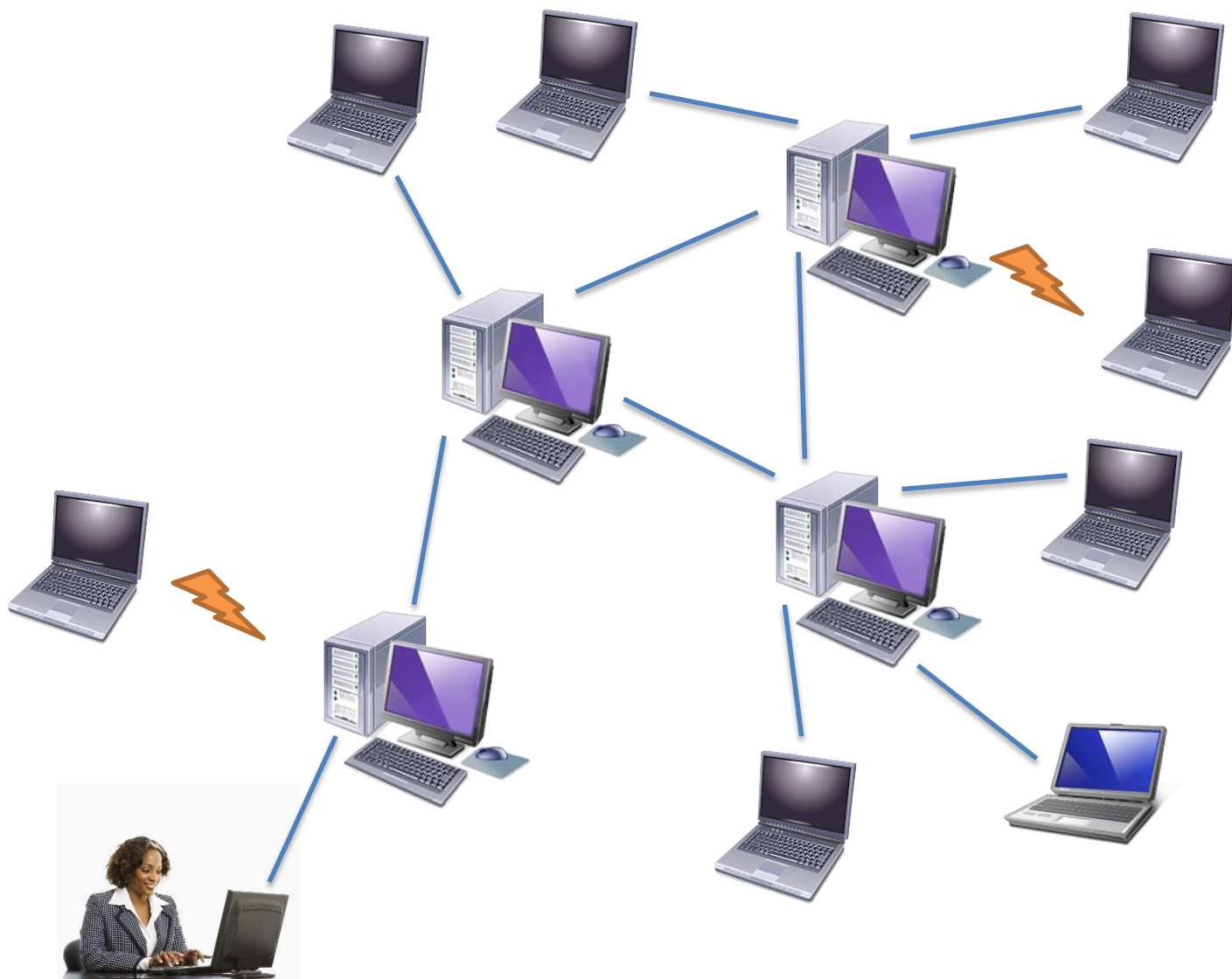
Основы алгоритмизации

План занятия

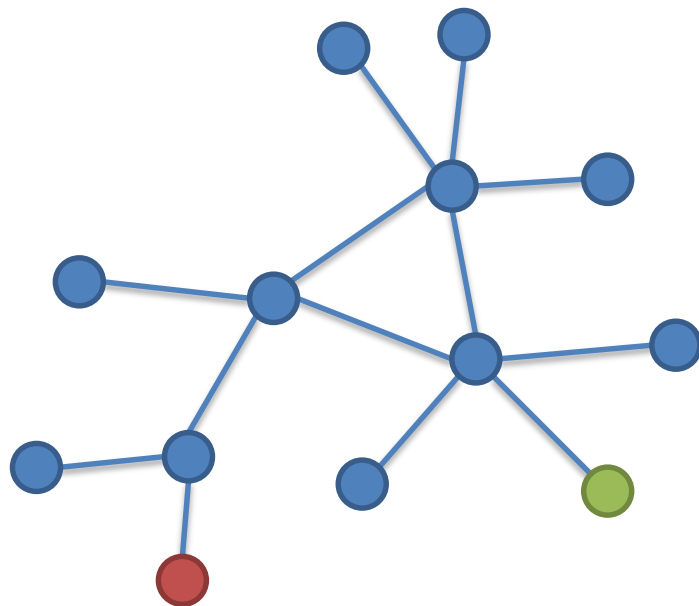
- Критерии сложности алгоритмов
- Уровни алгоритмической сложности
- Оценка времени работы алгоритма
- Примеры алгоритмов



Как работает интернет?



Как работает интернет?



«Граф Интернета»

Вершины:

- компьютеры;
- роутеры;

Связи:

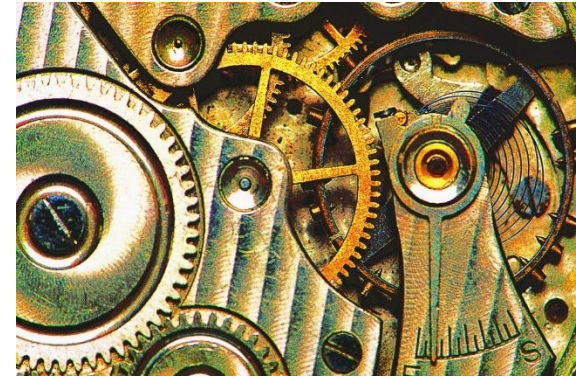
- соединения (провода, WiFi, ...)

Типовые алгоритмы:

- поиск в ширину;
- алгоритм Дейкстры;
- алгоритм Форда-Беллмана
- ...

Сложность алгоритмов

- Сложность восприятия
- Алгоритмическая сложность
- Временные затраты



Сложность восприятия

// Пример 1

```
int tmp = x;  
x = y;  
y = tmp;
```

// Пример 2

```
x += y;  
y = x - y;  
x -= y;
```

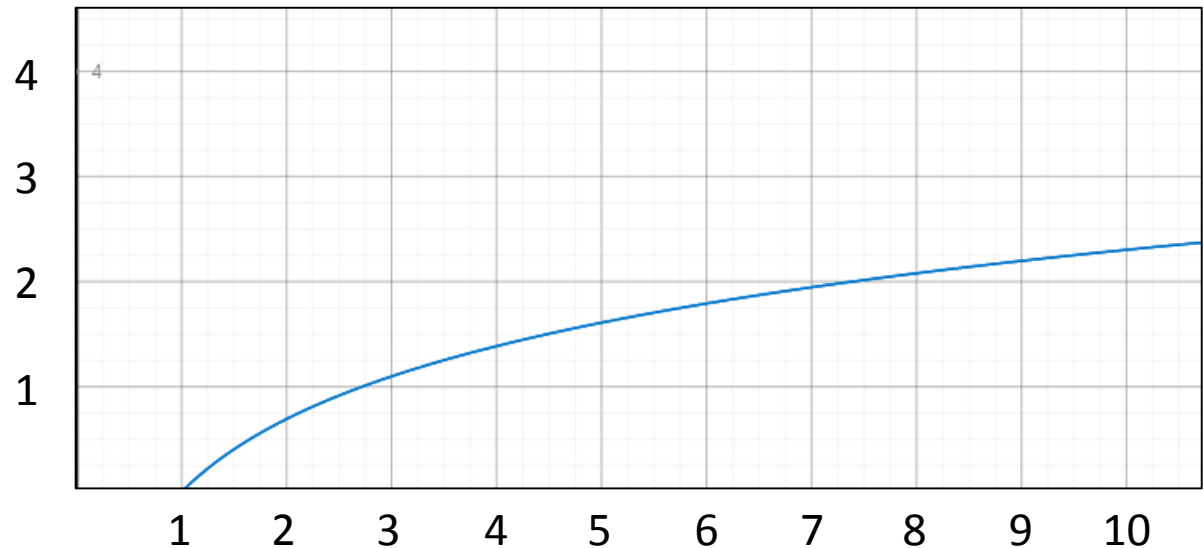
// Пример 3

```
x = x ^ y;  
y = x ^ y;  
x = x ^ y;
```

Какой из примеров проще для восприятия?

Алгоритмическая сложность

- $const$
- $\log(n)$
- n
- $n \cdot \log(n)$
- $n^{\text{число}}$
- число^n
- $n!$



Алгоритмическая сложность

- $const$

- $\log(n)$

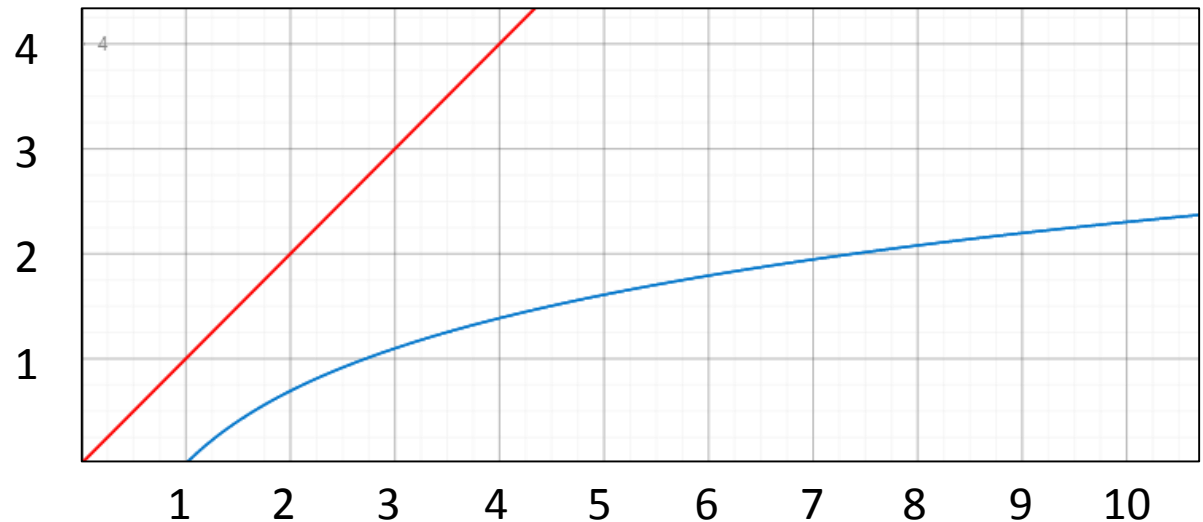
- n

- $n \cdot \log(n)$

- $n^{\text{число}}$

- число^n

- $n!$



Алгоритмическая сложность

- $const$

- $\log(n)$

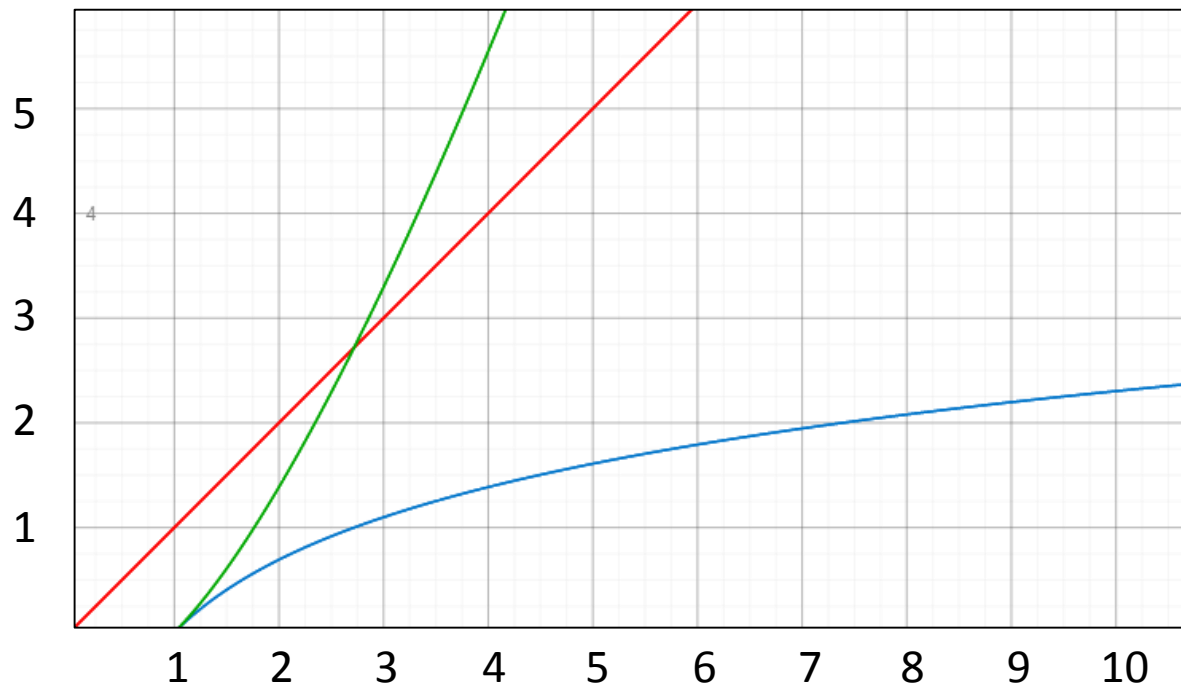
- n

- $n \cdot \log(n)$

- $n^{\text{число}}$

- число^n

- $n!$



Алгоритмическая сложность

- $const$

- $\log(n)$

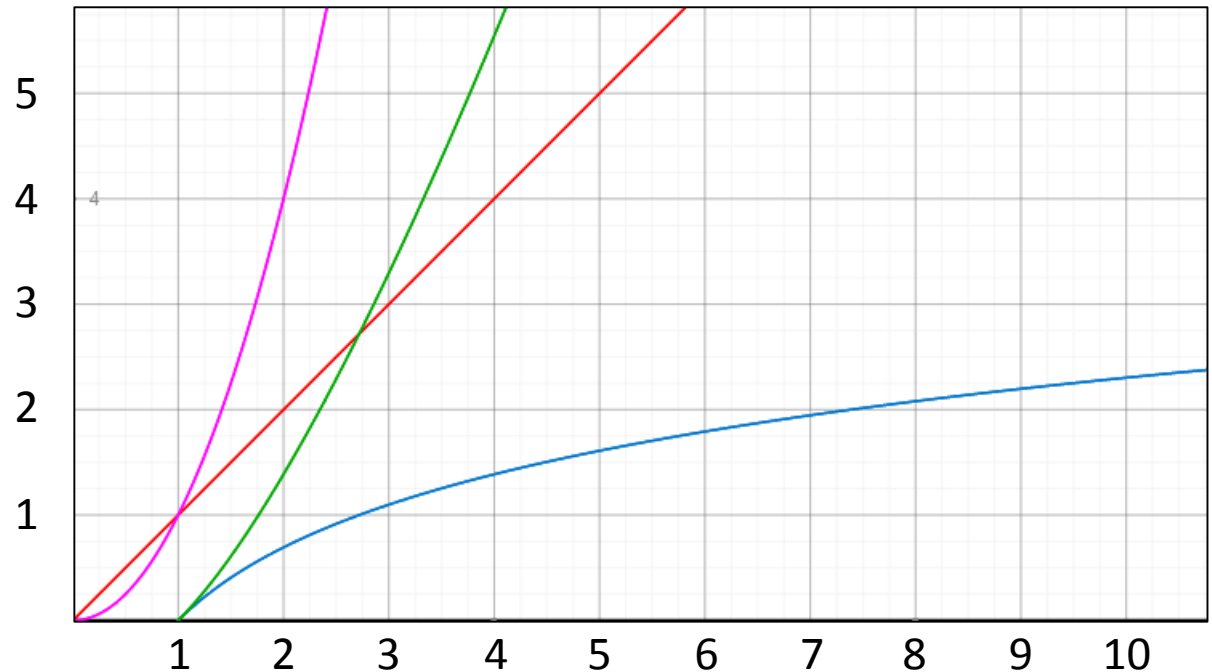
- n

- $n \cdot \log(n)$

- $n^{\text{число}}$

- число^n

- $n!$



Алгоритмическая сложность

- $const$

- $\log(n)$

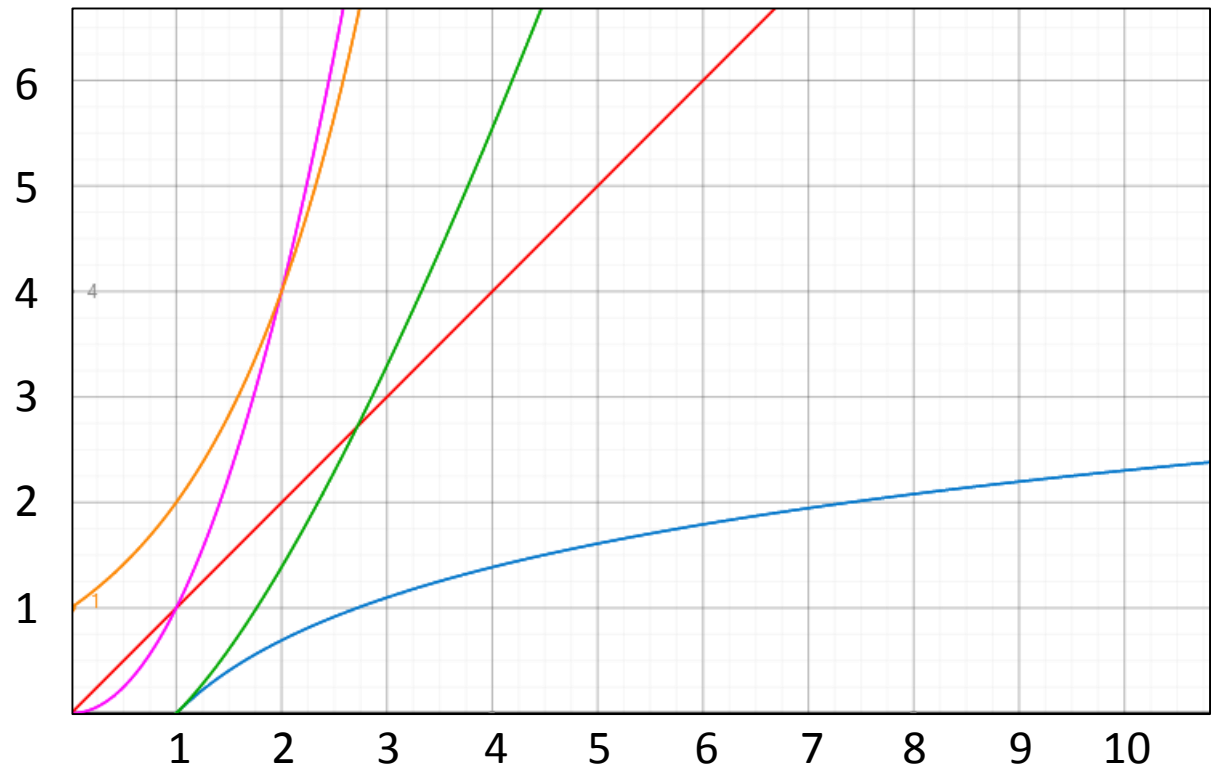
- n

- $n \cdot \log(n)$

- $n^{\text{число}}$

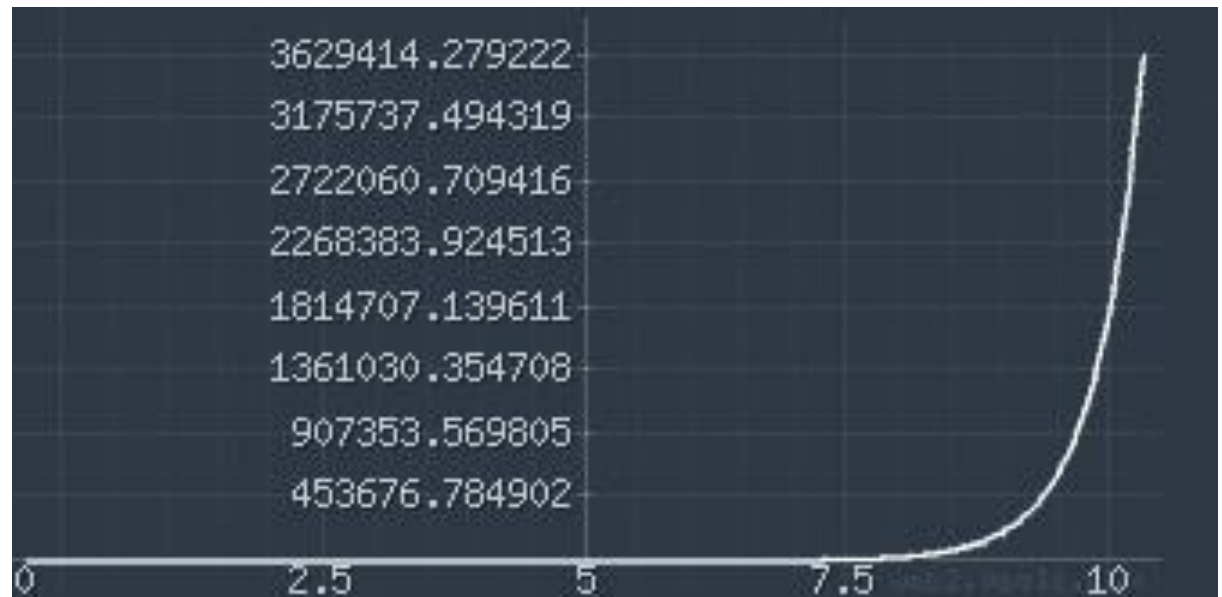
- число^n

- $n!$



Алгоритмическая сложность

- $const$
- $\log(n)$
- n
- $n \cdot \log(n)$
- $n^{\text{число}}$
- число^n
- ☐ $n!$



Оценка реального времени работы

```
var sw = new Stopwatch();

sw.Start();

int sum = 0;
for (int i = 1; i < 100; i++)
{
    sum += i;
}

sw.Stop();

Console.WriteLine(sw.ElapsedMilliseconds);
Console.WriteLine(sw.Elapsed.TotalMilliseconds);
Console.WriteLine(sw.ElapsedTicks);
```

Проблемы реальной оценки времени

1

Сторонние процессы

2

Компиляция на лету

3

Отладчик

4

Исходные данные

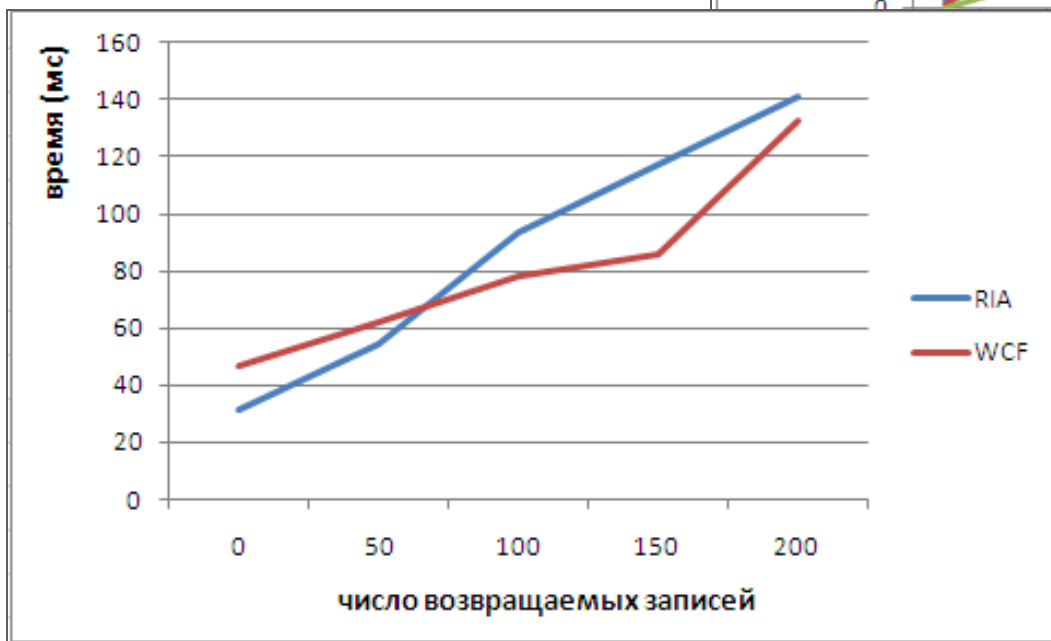
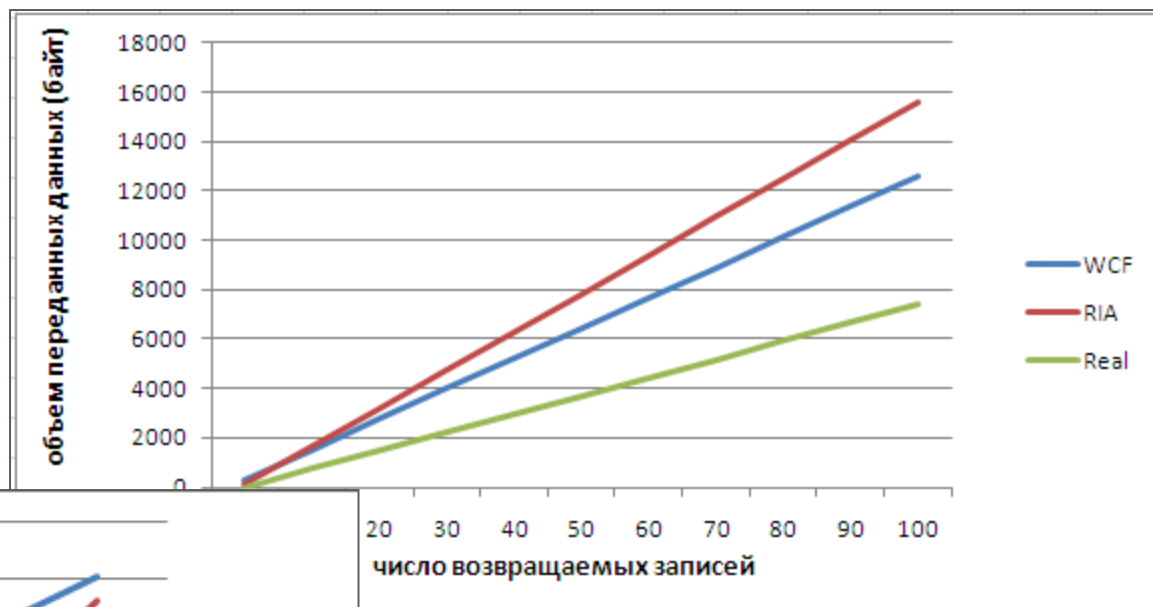
5

Случайность процесса

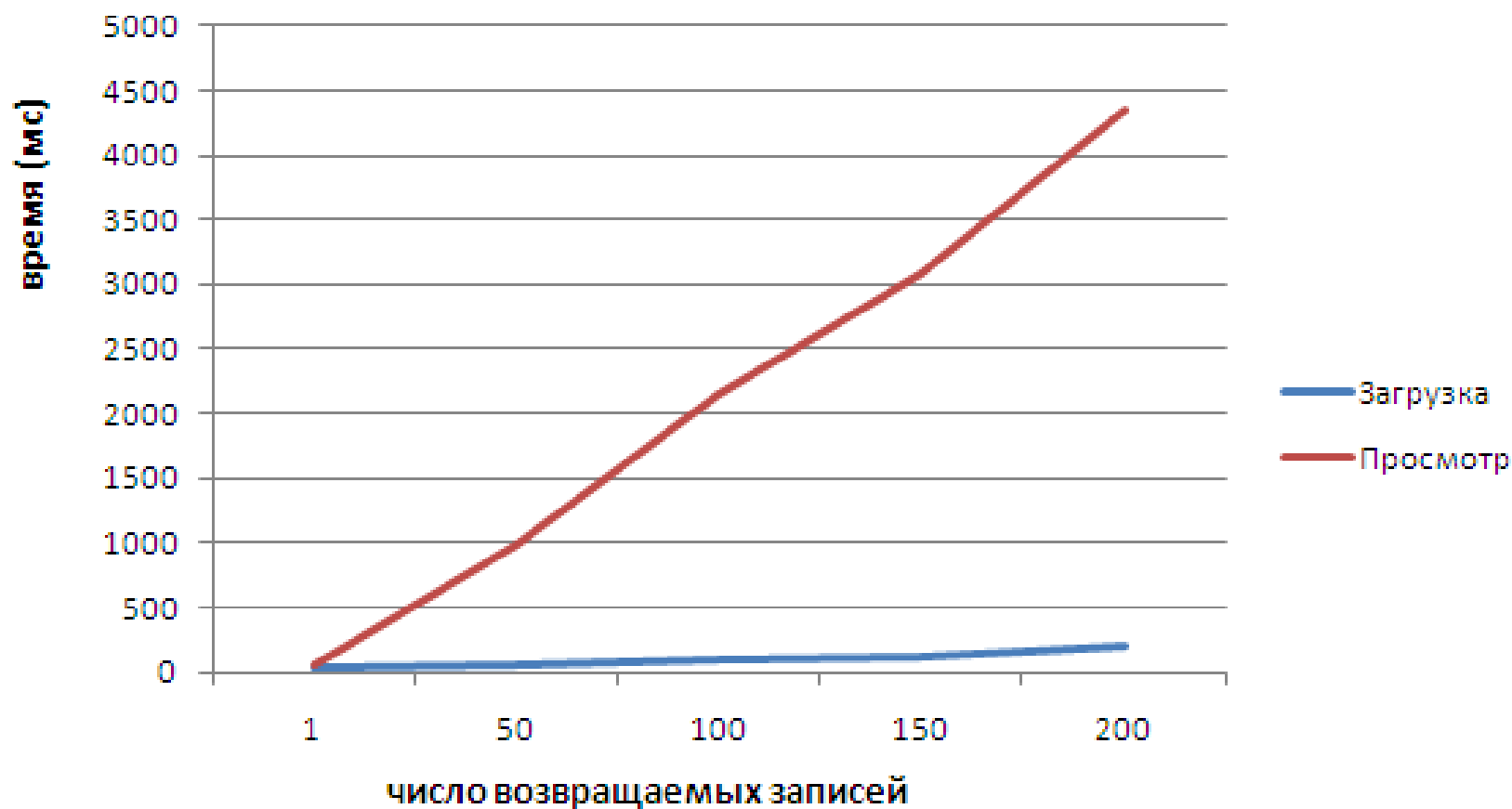
6

Программа — это не только ваш алгоритм

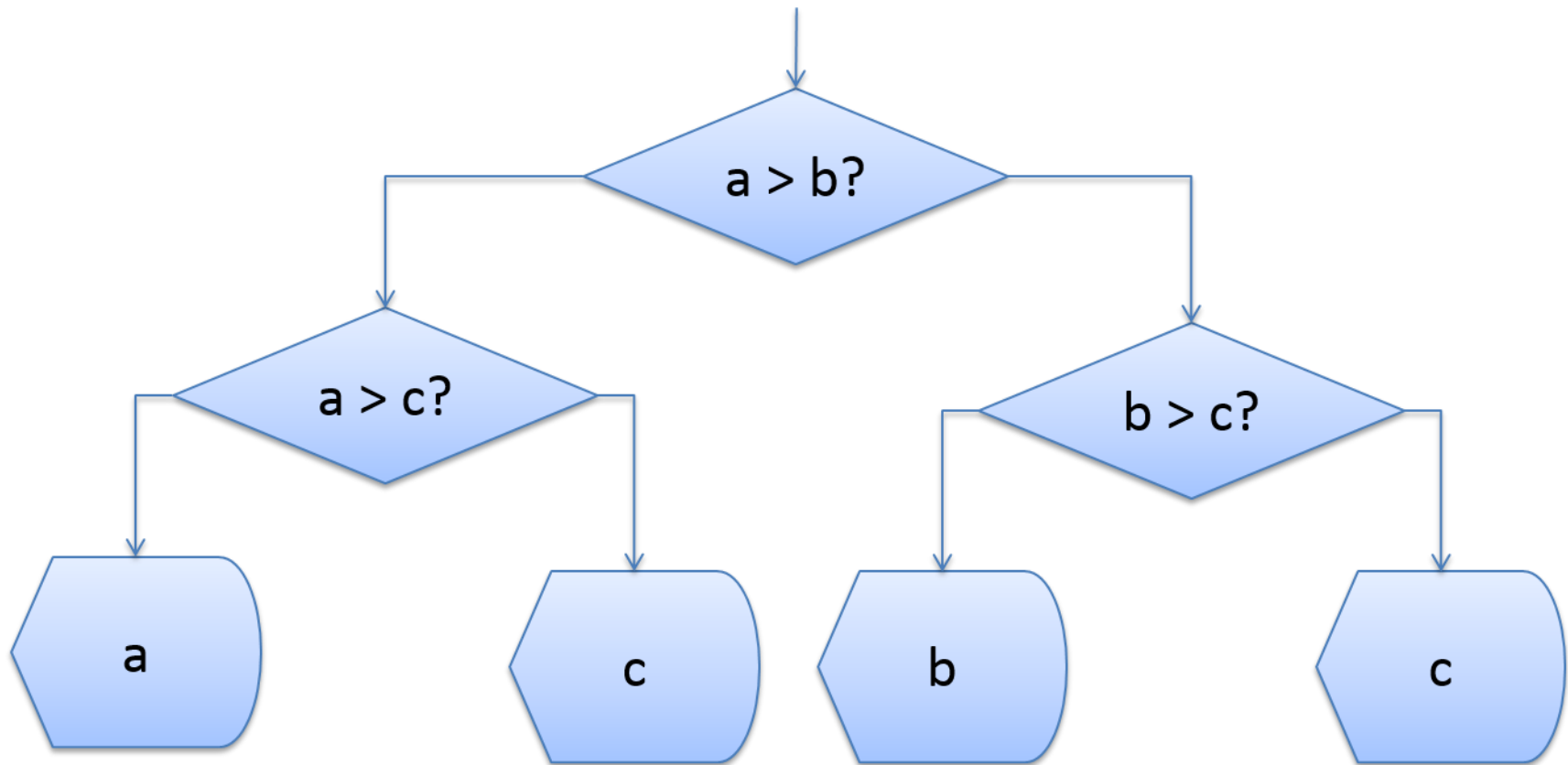
Оценка реального времени работы



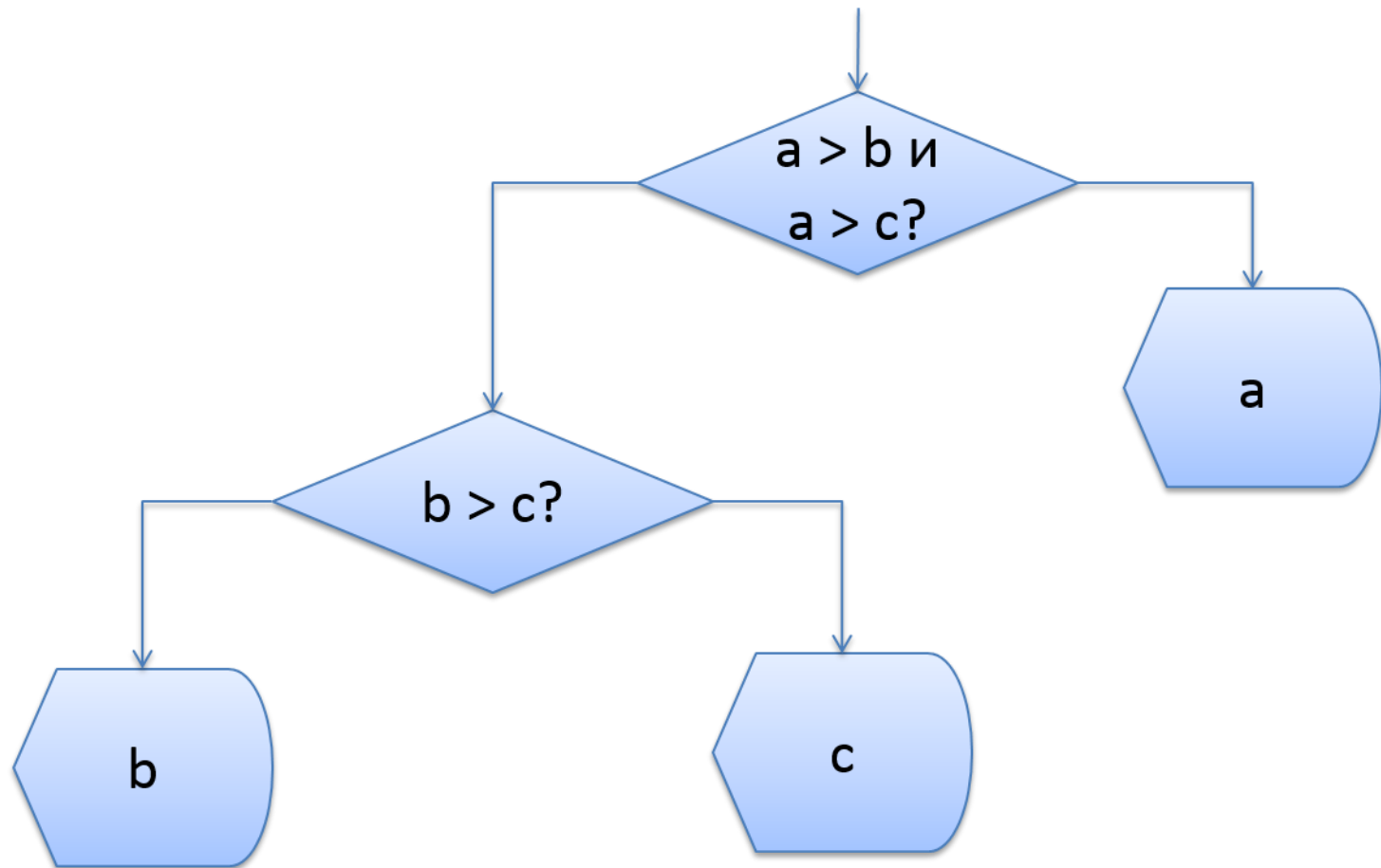
Оценка реального времени работы



Симметричный алгоритм



Асимметричный алгоритм



Решение конкретной задачи

```
public int MaximumGrowth(int[] array)
{
    int max = 0;

    for (int i = 0; i < array.Length; i++)
    {
        if (max < array[i])
        {
            max = array[i];
        }
    }

    return max;
}
```

Поиск максимума

```
public int Maximum(int[] array)
{
    int max = array[0];

    for (int i = 1; i < array.Length; i++)
    {
        if (max < array[i])
        {
            max = array[i];
        }
    }

    return max;
}
```

Типонезависимый метод

```
public T Maximum<T>(T[] array)
{
    T max = array[0];

    for (int i = 1; i < array.Length; i++)
    {
        if (max < array[i])
        {
            max = array[i];
        }
    }

    return max;
}
```

(parameter) T[] array

Error:

Operator '<' cannot be applied to operands of type 'T' and 'T'

Работающий типонезависимый метод

```
public T Maximum<T>(T[] array) where T : IComparable<T>
{
    T max = array[0];

    for (int i = 1; i < array.Length; i++)
    {
        if (max.CompareTo(array[i]) < 0)
        {
            max = array[i];
        }
    }

    return max;
}
```

Спасибо за внимание!

Контактная информация:

Дмитрий Верескун

Инструктор

EPAM Systems, Inc.

Адрес: Саратов, Рахова, 181

Email: Dmitry_Vereskun@epam.com

<http://www.epam.com>