Algorithms 2

TEL-RAN
by Starta Institute

Big-O обозначение

ПЛАН ЗАНЯТИЯ

1. Повторение изученного
2. Разбор домашнего задания
3. Теория: Оценка сложности алгоритма
4. Практическая работа
5. Оставшиеся вопросы

TEL-RAN
by Starta Institute

TEL-RAN
by Starta Institute

1 ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

TEL-RAN
by Starta Institute

Алгоритм - набор инструкций, описывающих последовательность действий для достижения конкретного результата.

Повторение

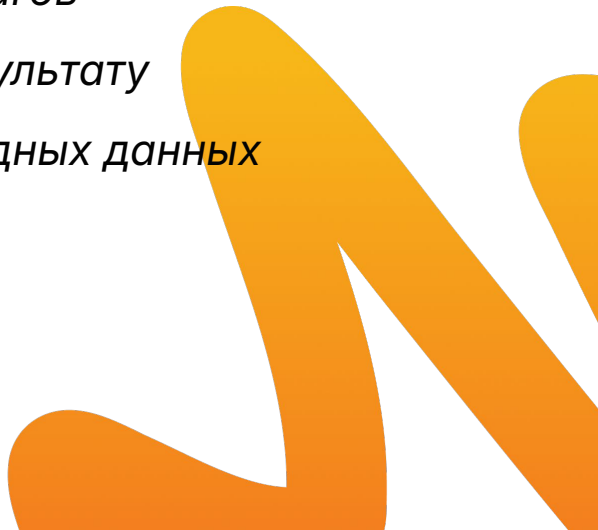
TEL-RAN
by Starta Institute

1. **Свойства алгоритма**
    a. Упорядоченность - ???
    b. Детерминированность
    c. Конечность
    d. Результативность
    e. Универсальность

Повторение

TEL-RAN
by Starta Institute

1. **Свойства алгоритма**
    a. Упорядоченность - *четкая последовательность действий*
    b. Детерминированность - *не двусмысленный*
    c. Конечность - *завершается за конечное число шагов*
    d. Результативность - *приводит к конкретному результату*
    e. Универсальность - *применим к множеству исходных данных*



Повторение

TEL-RAN
by Starta Institute

1. **Свойства алгоритма**
2. **Способы описания алгоритмов**
   a. Текстовый
   b. Графический (блок-схемы)
   c. Псевдокод

TEL-RAN
by Starta Institute

2 РАЗБОР ДОМАШНЕГО ЗАДАНИЯ

Алгоритм сложения 3 чисел

TEL-RAN
by Starta Institute

Задача: Найти наибольшее число из трех заданных чисел.

Алгоритм

Начало

- Получаем три числа для сравнения.
- Выбираем первое число в качестве текущего наибольшего.

Сравнение со вторым числом:

- Сравниваем второе число с текущим наибольшим.
- Если второе число больше, запоминаем его как новое наибольшее.

Сравнение с третьим числом:

- Сравниваем третье число с текущим наибольшим.
- Если третье число больше, запоминаем его как новое наибольшее.

Конец

TEL-RAN
by Starta Institute

3

Теория: Оценка сложности алгоритма

Сложность алгоритмов

TEL-RAN
by Starta Institute

В чем заключается сложность алгоритма?

Сложность алгоритмов

TEL-RAN
by Starta Institute

В чем заключается сложность алгоритма?

- Время выполнения
- Объем используемой памяти

Сложность алгоритмов

TEL-RAN
by Starta Institute

В чем заключается сложность алгоритма?

- Время выполнения
  - **Можно ли просто замерить время работы алгоритма?**
- Объем используемой памяти

Критерии сложности алгоритмов

TEL-RAN
by Starta Institute

Время выполнения:

- Измеряется в количестве операций
- Зависит от размера входных данных.
- Оценивается в худшем, среднем и лучшем случаях.

Объем используемой памяти:

- Измеряется в объеме памяти, необходимом для работы алгоритма.
- Зависит от размера входных данных.
- Оценивается в худшем, среднем и лучшем случаях.

Асимптотический анализ

TEL-RAN
by Starta Institute

Метод оценки поведения алгоритма при стремлении размера входных данных к бесконечности. Он фокусируется на скорости роста времени выполнения или объема используемой памяти алгоритмом.

Результатом асимптотического анализа является оценка вида **O(f(n))**, где f(n) - некоторая функция, описывающая порядок роста времени или памяти с увеличением n (размера входных данных).

$$O(f(n))$$

Порядок роста

TEL-RAN
by Starta Institute

Порядок роста описывает то, как сложность алгоритма растет с увеличением размера входных данных. Порядок роста представляется в виде O-нотации:

O(f(x)), где f(x) — формула, выражающая сложность алгоритма.

O(1) – **Константный**

Порядок роста O(1) означает, что вычислительная сложность алгоритма не зависит от размера входных данных.

```
public int getSize(int[] arr) {
    return arr.length;
}
```

Порядок роста

TEL-RAN
by Starta Institute

O(n) **– линейный**

Порядок роста O(n) означает, что сложность алгоритма линейно растет с увеличением входного массива.

Если линейный алгоритм обрабатывает один элемент 1 секунду, то сто элементов обработается за сто секунд.

```
public long getSum(int[] arr) {
      long sum = 0;
      for (int i = 0; i < arr.length; i++) {
          sum += i; }
     return sum;
 }
```

Порядок роста

TEL-RAN
by Starta Institute

O(log n) **– логарифмический**

Порядок роста O(log n) означает, что время выполнения алгоритма растет логарифмически с увеличением размера входного массива.

Большинство алгоритмов, работающих по принципу «деления пополам», имеют логарифмическую сложность.

Пример:

Алгоритм двоичного поиска

Порядок роста

TEL-RAN
by Starta Institute

O(n log n) **– линейно-логарифмический**

Некоторые алгоритмы типа «разделяй и властвуй» попадают в эту категорию.

Пример:

Сортировка слиянием и быстрая сортировка

Порядок роста

TEL-RAN
by Starta Institute

O(n2) **– квадратичный**

Время работы алгоритма O(n^2) зависит от квадрата размера входного массива.

Квадратичная сложность — повод задуматься и переписать алгоритм.

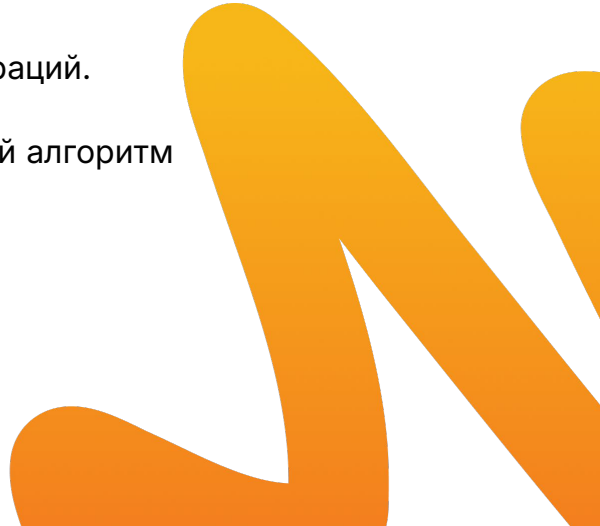
Массив из 100 элементов потребует 10000 операций

Массив из миллиона элементов потребует 1 000 000 000 000 (триллион) операций.

Если одна операция занимает **миллисекунду** для выполнения, квадратичный алгоритм будет обрабатывать миллион элементов **32 года**.

Пример:

Алгоритм пузырьковая сортировка

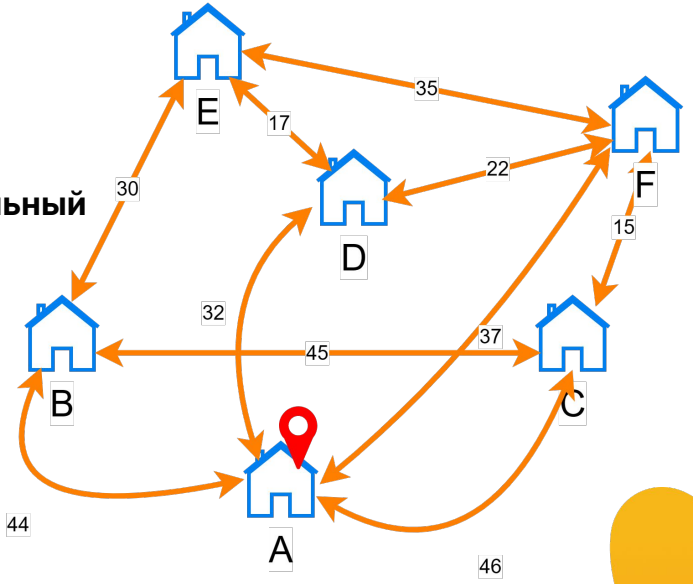


Порядок роста

O(n!) **– факториальный**

Очень медленный алгоритм.

Пример:



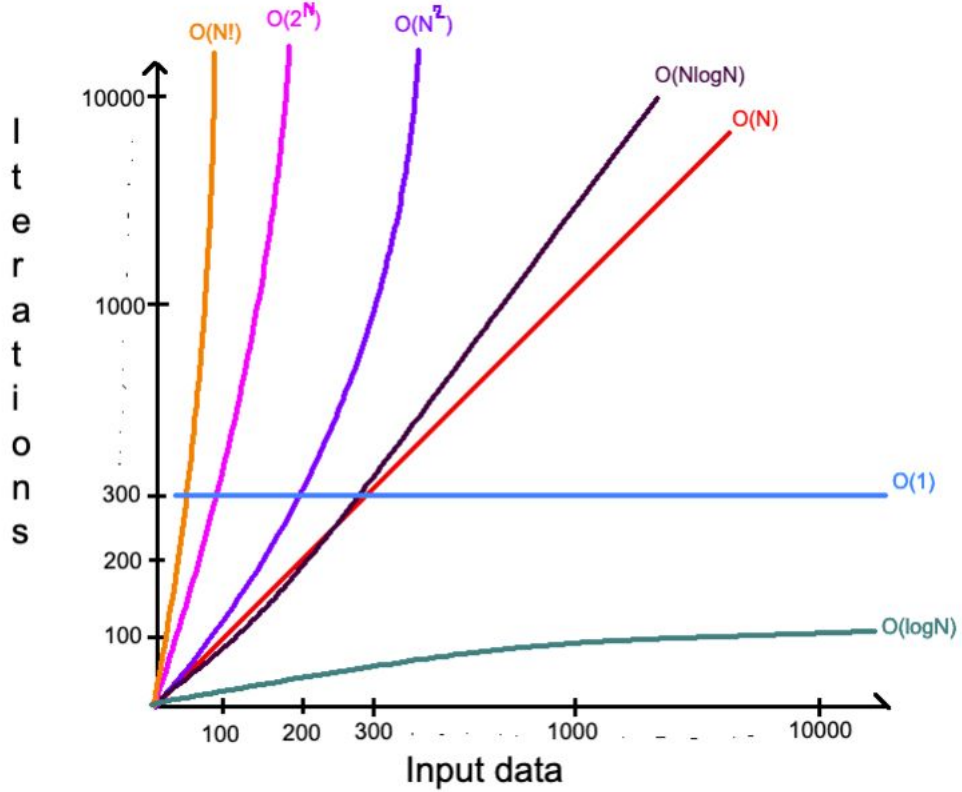
Задача коммивояжёра

Коммивояжер должен посетить n городов, побывав в каждом ровно один раз, и вернуться в исходный город, при этом минимизировав суммарную длину маршрута.

Для **15 городов** существует **43 миллиарда** маршрутов, а для **18 городов** уже **177 триллионов**.

Порядок роста

TEL-RAN
by Starta Institute



Наилучший, средний и наихудший

TEL-RAN
by Starta Institute

- **Наихудший случай** показывает нам гарантированную верхнюю границу времени выполнения алгоритма. Это важно для критически важных систем, где недопустимы задержки.
- **Средний случай** дает нам представление о том, как алгоритм будет работать в большинстве ситуаций.
- **Наилучший случай** может быть полезен для понимания теоретических возможностей алгоритма, но на практике он встречается редко.

Наилучший, средний и наихудший

TEL-RAN
by Starta Institute

Как это используется?

При выборе алгоритма для решения конкретной задачи мы должны учитывать не только его сложность в наихудшем случае, но и то, как часто этот наихудший случай может встречаться на практике.

Что мы в итоге измеряем и всегда ли это работает?

TEL-RAN
by Starta Institute

Асимптотический анализ не идеален, но это лучший доступный способ анализа алгоритмов.

Два алгоритма сортировки, которые занимают на машине:

1000 nLogn и **2 nLogn**.

Мы не можем судить, какой из них лучше, поскольку мы игнорируем константы.

Таким образом, вы можете в конечном итоге выбрать алгоритм, который асимптотически медленнее, но быстрее для вашего программного обеспечения.

Важно:

TEL-RAN
by Starta Institute

- Скорость алгоритма измеряется не в секундах, а в приросте количества операций.
- Насколько быстро возрастает время работы алгоритма в зависимости от увеличения объема входящих данных.
- Время работы алгоритма выражается при помощи нотации большого «O».
- Алгоритм со скоростью O(log n) быстрее, чем со скоростью O(n), но он становится намного быстрее по мере увеличения списка элементов.

Экспресс-опрос

TEL-RAN
by Starta Institute

- **Вопрос 1.**

  Что обозначает Big O?

- **Вопрос 2.**

  Почему игнорируются константы?

TEL-RAN
by Starta Institute

4 ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Практическое задание

TEL-RAN
by Starta Institute

Работаем с:

org.telran.**lecture_02_big_o**

- Задачи выполняем в соответствии с “**рекомендуемый_порядок.md**”
- Результаты в виде **PullRequest**

Полезные ссылки

TEL-RAN
by Starta Institute

- Задача коммивояжёра — Википедия

ЗАКЛЮЧЕНИЕ

TEL-RAN
by Starta Institute