

Алгоритмы поиска. Диаграммы Эйлера- Венна

На уроке мы узнаем

- Ознакомимся с алгоритмами поиска
- Способы визуализации множеств с использованием диаграмм Эйлера-Венна

Что такое массивы данных?

Ответ

Массив — это структура данных, которая представляет собой набор элементов одного типа, расположенных в памяти последовательно. Каждый элемент массива имеет индекс, с помощью которого к нему можно обратиться.

Пример

- Массив чисел: `[2, 4, 6, 8, 10]`
- Массив строк: `["apple", "banana", "cherry"]`
- Массив оценок учеников: `[10, 8, 6, 9, 7]`

Какие методы поиска вы знаете?

Линейный поиск

- Элементы массива проверяются последовательно, пока не будет найдено искомое значение или пока не закончится массив.
- **Пример:** искать число 5 в массиве [1, 2, 3, 5, 7]

Бинарный поиск

- Используется в **отсортированных** массивах. Массив делится пополам, и проверяется средний элемент. Если искомое значение меньше среднего, поиск продолжается в левой части, иначе — в правой.
- **Пример:** искать число 5 в отсортированном массиве [1, 3, 5, 7, 9]

Что такое множества и их пересечение?

Ответ

- **Множество:** это структура данных, которая представляет собой набор уникальных элементов.
- **Пересечение множеств:** это элементы, которые присутствуют одновременно в обоих множествах.

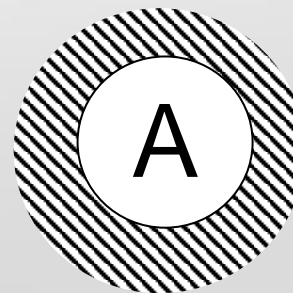
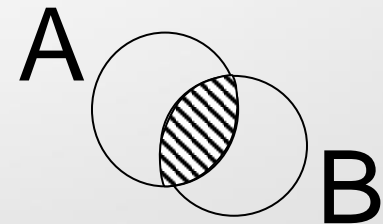
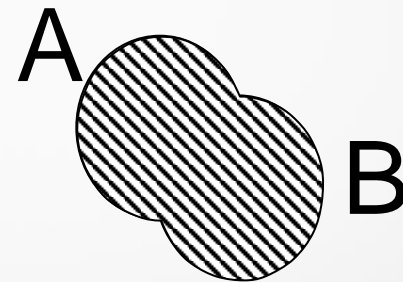
Пример

- Множество A: {1, 2, 3, 4}
- Множество B: {3, 4, 5, 6}
- Пересечение $A \cap B$: {3, 4}

Диаграммы Венна (круги Эйлера)

Диаграммы Эйлера-Венна помогают наглядно представить отношения между множествами:

- **Объединение множеств ($A \cup B / A+B$):** включает все элементы из A и B .
- **Пересечение множеств ($A \cap B / A \cdot B$):** показывает общие элементы.
- **Инверсия множества ($\neg A / \bar{A}$):** включает все элементы, которые не принадлежат данному множеству. Например, инверсия A ($\neg A$) — это элементы, отсутствующие в множестве A .



Связь между множествами, диаграммами и алгоритмами поиска

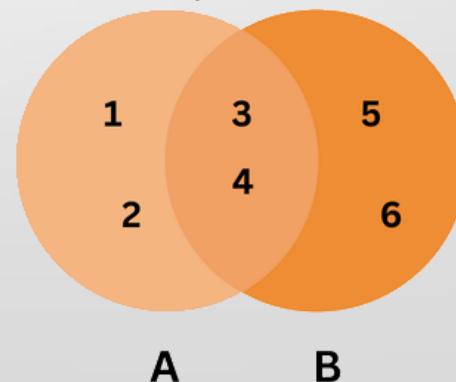
Алгоритмы поиска могут быть использованы для работы с множествами:

- **Линейный поиск** помогает проверить, принадлежит ли элемент множеству.
- **Бинарный поиск** ускоряет проверку в отсортированных массивах, которые могут представлять множества.

Задача: Найти общие элементы двух массивов $A = \{1, 2, 3, 4\}$ и $B = \{3, 4, 5, 6\}$.

Алгоритм:

- Используем линейный поиск для сравнения каждого элемента массива A с элементами массива B .
- В результате пересечение $A \cap B = \{3, 4\}$



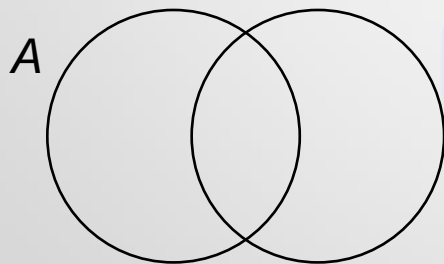
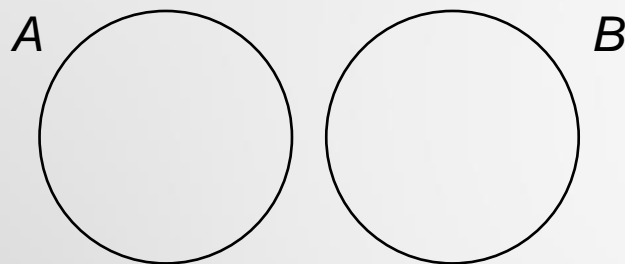
Задачи

Известно количество сайтов, которых находит поисковый сервер по следующим запросам:

<i>Запрос</i>	<i>Количество сайтов</i>
<i>огурцы</i>	<i>100</i>
<i>помидоры</i>	<i>200</i>
<i>огурцы & помидоры</i>	<i>50</i>

Сколько сайтов будет найдено по запросу
огурцы | помидоры

Решение



B

50

огурцы & помидоры

$$N_{A/B} = N_A + N_B - N_{A\&B}$$

огурцы | помидоры

250

огурцы

100

помидоры

200

$$N_{A/B} =$$

<https://github.com/agorbatniov/inf10>

Алгоритмы поиска: Линейный

- Последовательная проверка элементов массива.
- Пример: найти заданное число в массиве.

Псевдокод

```
Ввод: массив A, искомое значение x
Для каждого элемента e в A:
    Если e равно x:
        Вывести "Найдено" и завершить
Если x не найдено:
    Вывести "Не найдено"
```

Алгоритмы поиска: Бинарный

Псевдокод

- Применяется к отсортированным массивам.
- Пример: поиск числа в массиве делением пополам.

<https://github.com/agorbatniov/inf10>

```
Ввод: отсортированный массив A, искомое значение x
Левый = 0, Правый = длина(A) - 1
Пока Левый <= Правый:
    Средний = (Левый + Правый) // 2
    Если A[Средний] == x:
        Вывести "Найдено" и завершить
    Иначе, если A[Средний] > x:
        Правый = Средний - 1
    Иначе:
        Левый = Средний + 1
Вывести "Не найдено"
```

Практическая работа

Часть 1: Алгоритмы поиска

1. Написать программу линейного поиска числа в массиве на Python.
2. Написать программу бинарного поиска числа в массиве.

Практическая работа

Часть 2: Диаграммы Эйлера-Венна

1. Нарисовать диаграмму для множества $A = \{3, 4, 5, 6\}$ и $B = \{5, 6, 7, 8\}$.
2. Найти пересечение, объединение и инверсию множеств.

Какой алгоритм поиска быстрее для больших массивов? Почему?

Ответ:

Бинарный поиск многократно делит массив пополам, что позволяет сократить количество проверок. Однако он работает только с отсортированными массивами.

Как используются диаграммы Эйлера-Венна для анализа данных?

Ответ:

Диаграммы Эйлера-Венна используются для наглядного представления отношений между множествами. Они позволяют:

- Выявлять общие элементы (пересечения) между наборами данных.
- Анализировать уникальные элементы каждого множества.
- Представлять объединение или инверсию данных для анализа связей.

Что мы сегодня выучили

1. Алгоритмы поиска:

- Линейный
- Бинарный

2. Способы визуализации множеств с использованием диаграмм Эйлера-Венна

Рефлексия

- Что понравилось на уроке?
- Что осталось непонятным?