Сортировка и поиск

AIT
Hi-tech team

НАШИ ПРАВИЛА

Включенная камера

Вопросы по поднятой руке

Не перебиваем друг друга

Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору

Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Сформировать понимание базовых алгоритмов сортировок и поиска. Научиться применять их на практике.

ПЛАН ЗАНЯТИЯ

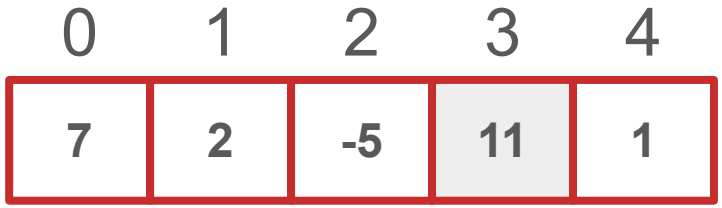
- Задача поиска, линейный поиск
- Бинарный поиск
- Сортировка выбором

ЗАДАЧА ПОИСКА, ЛИНЕЙНЫЙ ПОИСК

Поиск - действие, направленное на проверку наличия элемента в наборе данных, либо определяющее конкретную позицию этого элемента

Линейный поиск - алгоритм поиска, заключающийся в последовательном переборе элементов набора на предмет их соответствия искомому

В **худшем случае** (поиск последнего вхождения элемента при его отсутствии в коллекции) алгоритму придется пройти и проверить **все** элементы коллекции. На больших данных это может привести к медленной работе функции поиска.



Поиск числа **11** в зависимости от реализации вернет либо **true** (присутствует), либо **3** (позиция элемента)

Поиск числа **777** в зависимости от реализации вернет либо **false** (отсутствует), либо **-1** (невозможный индекс). При этом будут проверены **все** элементы.

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Задача поиска

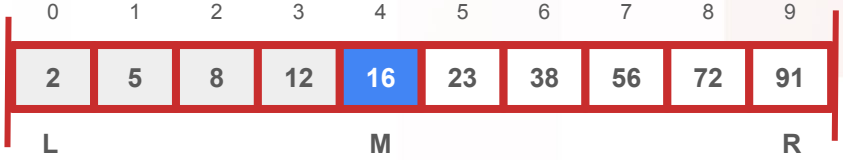
Линейный поиск

“Пробег” всех элементов в худшем случае

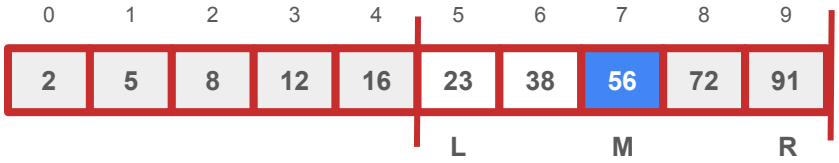
БИНАРНЫЙ ПОИСК

Бинарный поиск - алгоритм поиска, который можно применить на **упорядоченных** данных. Суть алгоритма аналогична поиску слова в словаре, где слова упорядочены по алфавиту.

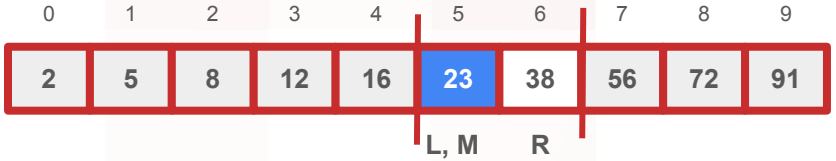
Шаг 1:
левый курсор **L** на позиции **0**
правый курсор **R** на позиции **9**
середина **M** - элемент **16** под индексом **L + (R - L) / 2 = 4**,
при этом мы знаем, что **23 > 16**, следовательно, смотрим **правую** половину:



Шаг 2:
левый курсор **L** на позиции **M + 1 = 5**
правый курсор **R** на той же позиции **9**
середина **M** - элемент **56** под индексом **L + (R - L) / 2 = 7**,
при этом мы знаем, что **23 < 56**, следовательно, смотрим левую половину:



Шаг 3:
левый курсор **L** на той же позиции **5**
правый курсор **R** на позиции **M - 1 = 6**
середина **M** - элемент **23** под индексом **L + (R - L) / 2 = 5**,
элемент 23 **найден**



Следует заметить, что потребовалось выполнить всего 3 сравнения

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

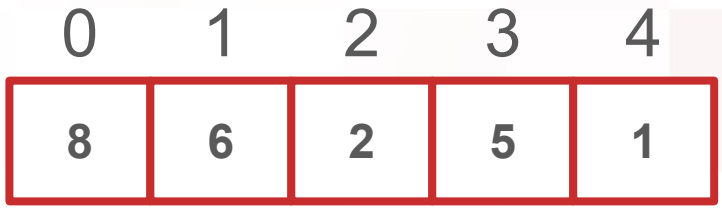
Условия работы бинарного поиска

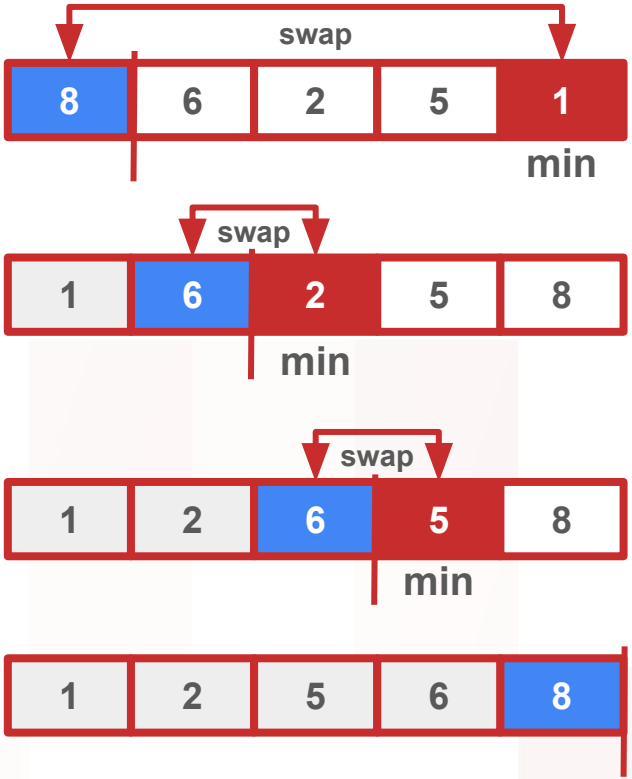
Меньше, чем при линейном поиске, количество операций сравнения

СОРТИРОВКА ВЫБОРОМ

Сортировка - процесс, позволяющий **упорядочить** набор элементов по определенному критерию. Сортировка **ускоряет процесс поиска**.

Существует большое количество алгоритмов сортировки, отличающихся скоростью работы. Рассмотрим простейший алгоритм сортировки выбором (**selection sort**)





СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Для чего нужна сортировка?

Алгоритм и реализация сортировки выбором

ПОИГРАЕМ ;)

- Поиск
- Сортировка выбором
- Бинарный поиск
- Худший случай бинарного поиска

ДОМАШНЕЕ ЗАДАНИЕ

Либо это:

Расписать словами шаги действий по применению одного из рассмотренных алгоритмов для массива `int[] data = {5,5,7,3,2};` **если выбрали бинарный поиск, то можете не расписывать этап сортировки.**
По желанию: сделать вышесказанное для всех рассмотренных алгоритмов.

Либо это:

Написать программу, которая реализует один из рассмотренных алгоритмов.
По желанию: написать реализацию для всез алгоритмов.

AIT
Hi-tech team

Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH

AIT
Hi-tech team