



Программирование на Python

Презентация занятия

Алгоритмы и структуры данных. Часть 1.

10 занятие



инжинириум[®]

МГТУ им. Н.Э. Баумана

2019

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ. ОРГАНИЗАЦИОННАЯ ИНФОРМАЦИЯ

- Тема занятия
- Цели и задачи занятия
- Результаты занятия
- Материалы для преподавателя
- Материалы для ученика
- Тайминг проведения занятия

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

- Алгоритмы
- Классы алгоритмов
- Семейство алгоритмов сортировки

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

- Обзор базовых алгоритмов
- Оценка временной сложности
- Подходы к оптимизации кода
- Эффективность кода



ВВЕДЕНИЕ. ОРГАНИЗАЦИОННАЯ ИНФОРМАЦИЯ

Тема: Алгоритмы и структуры данных. Часть 1.

Цели и задачи:

- Рассказать об определении алгоритма.
- Рассказать о современных подходах к оптимизации кода.
- Рассказать про общую алгоритмическую сложность.
- Описать основные классы алгоритмов.
- Рассмотреть базовые алгоритмы сортировки.
- Рассказать про оценку временной сложности.

По результатам занятия слушатель будет знать:

- Что такое алгоритмы и для чего их использовать.
- Какие существуют классы алгоритмов.
- Какие есть виды алгоритмов сортировки.





Тема: Алгоритмы и структуры данных. Часть 1.

По результатам занятия слушатель будет уметь:

- Использовать базовые алгоритмы в программном коде.
- Работать с оценкой сложности и видоизменять структуру классических алгоритмов.
- Оптимизировать и увеличивать эффективность программного кода.

Тайминг занятия

Таб.1

№	Этапы	время	Сумма
1	Алгоритмы	15 мин.	15
2	Базовые алгоритмы	30 мин.	30 мин.
3	<i>Перерыв</i>	5 мин.	5 мин.
4	Базовые алгоритмы	10 мин.	10 мин.
5	Сложность	30 мин.	30 мин.



Тема: Алгоритмы и структуры данных. Часть 1.

1. Алгоритмы

1.1 Что такое алгоритмы?

Алгоритмы - конечная совокупность точно заданных правил решения определенных задач, приводящие к определенному и однозначному результату.

1.2 Для чего они нужны?

Алгоритмы необходимы для увеличения эффективности программного кода : уменьшения времени выполнения, уменьшения затрат памяти.

1.3 Почему их необходимо использовать.

Применение алгоритмов в программном коде позволяет коду решать более общую задачу без потери эффективности. Стоит заметить, что для решения одной и той же задачи могут быть пригодны несколько видов алгоритмов. Главной задачей программиста является классификация задачи и выбор наиболее подходящего для данной задачи алгоритма.



Тема: Алгоритмы и структуры данных. Часть 1.

2. Базовые алгоритмы

2.1 Алгоритмы сортировки - это алгоритм для упорядочивания элементов в хранилище. Алгоритмы сортировки - отдельный класс алгоритмов, решающий исключительно одну задачу - задачу упорядочивания элементов в хранилище.

2.2 Почему нам нужно сортировать объекты

Упорядоченные наборы элементов (как следствие своей упорядоченности) позволяют находить наименьший или наибольший элемент в списке, выделять упорядоченные подпоследовательности. Основная идея - изменить порядок элементов в хранилище по определенному правилу сравнений элементов друг с другом.



Тема: Алгоритмы и структуры данных. Часть 1.

2.3 Реализация алгоритмов сортировки

2.3.1 Пузырек

Сортировка пузырьком - простейший из семейства алгоритмов сортировки. Эффективен только для небольших массивов.

Сортировка пузырьком - это метод сортировки массивов и списков путем последовательного сравнения и обмена соседних элементов, если предшествующий оказывается больше последующего.

В сортировке методом пузырька количество итераций внешнего цикла определяется длиной списка минус единица, так как когда второй элемент становится на свое место, то первый уже однозначно минимальный и находится на своем месте.



Тема: Алгоритмы и структуры данных. Часть 1.

Пусть имеется список [6, 12, 4, 3, 8].

За первую итерацию внешнего цикла число 12 переместится в конец. Для этого потребуется 4 сравнения во внутреннем цикле:

6 > 12? Нет

12 > 4? Да. Меняем местами

12 > 3? Да. Меняем местами

12 > 8? Да. Меняем местами

Результат: [6, 4, 3, 8, 12]

Далее попробуйте отсортировать список самостоятельно

Результат: [3, 4, 6, 8, 12]



Тема: Алгоритмы и структуры данных. Часть 1.

2.3.2 Мердж

Сортировка слиянием (merge sort) - алгоритм сортировки, основанный на принципе "разделяй и властвуй". Сначала главная задача разбивается на несколько подзадач меньшего размера. Затем эти подзадачи разбиваются на подподзадачи и так далее. В самом конце, решения элементарных подзадач комбинируются и получается исходное решение задачи.



Тема: Алгоритмы и структуры данных. Часть 1.

2.3.3 Квиксорт

Быстрая сортировка (quick sort) - один из самых быстрых и широкоизвестных алгоритмов сортировки, использующийся в промышленном программировании с некоторыми доработками. Основан на принципах bubble sort, но имеет ряд принципиальных отличий. В первую очередь производятся перестановки на наибольшем возможном расстоянии и после каждого прохода (шага сортировки) элементы делятся на пары независимых групп.



Тема: Алгоритмы и структуры данных. Часть 1.

3. Сложность

3.1 Об оценке эффективности алгоритма

Эффективность алгоритма - свойство алгоритма, связанное с вычислительными ресурсами при использовании алгоритма. Чаще всего это время выполнения и память. Оценивать эффективность необходимо для того, чтобы доказать, что "алгоритм А справляется с задачей лучше, чем алгоритм Б".

3.2 Затраты по памяти

Оценка эффективности по памяти - как много рабочей памяти (RAM) нужно для алгоритма. Нюансы: количество памяти для кода и количество памяти для данных, с которыми код работает. Оценка по памяти - используется редко, т.к. сильно зависит от машины.





Тема: Алгоритмы и структуры данных. Часть 1.

3.3 Затраты по времени

Оценка по времени (временная оценка) - как долго алгоритм занимает процессор. Удобен тем, что можно использовать итерационное время. Позволяет математизировать процесс оценки эффективности.

