

Python

2-месяц 7-урок

Тема: Структуры данных и алгоритмы

Что такое структуры данных?

По сути, это способы хранить и организовывать данные, чтобы эффективно решать различные задачи. Данные можно представить по-разному. В зависимости от того, что это за данные и что вы собираетесь с ними делать, одно представление подойдет лучше других.

Чтобы понять, почему так происходит, сперва поговорим об алгоритмах.

Алгоритмы

Алгоритм — такое хитроумное название для последовательности совершаемых действий.

Структуры данных реализованы с помощью алгоритмов, алгоритмы — с помощью структур данных. Всё состоит из структур данных и алгоритмов, вплоть до уровня, на котором бегают микроскопические человечки с перфокартами и заставляют компьютер работать. (Ну да, у Интела в услужении микроскопические люди. Поднимайся, народ!)

Любая данная задача реализуется бесконечным количеством способов. Как следствие, для решения распространённых задач изобрели множество различных алгоритмов.

Например, для сортировки неупорядоченного множества элементов существует до смешного большое количество алгоритмов: Сортировка вставками, Сортировка выбором, Сортировка слиянием, Сортировка пузырьком, Сортировка кучи, Быстрая сортировка, Сортировка Шелла, Сортировка Тима, Блочная сортировка, Поразрядная сортировка...

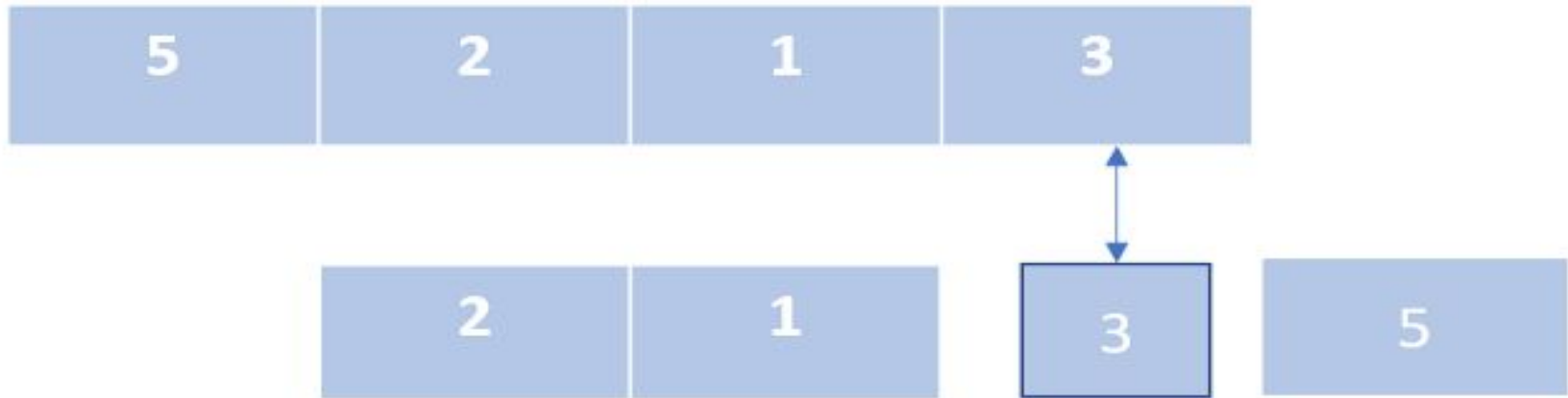
Некоторые из них значительно быстрее остальных. Другие занимают меньше памяти. Третьи легко реализовать. Четвёртые построены на допущениях относительно наборов данных. Каждая из сортировок подходит лучше других для определённой задачи. Поэтому вам надо будет сперва решить, какие у вас потребности и критерии, чтобы понять, как сравнивать алгоритмы между собой.

Быстрая сортировка (quick sort)

В этом алгоритме мы разбиваем список при помощи опорного элемента, сортируя значения вокруг него.

В нашей реализации мы выбрали опорным элементом последний элемент массива. Наилучшая производительность достигается тогда, когда опорный элемент делит список примерно пополам.

Интерактив



Сортировка пузырьком (bubble sort)

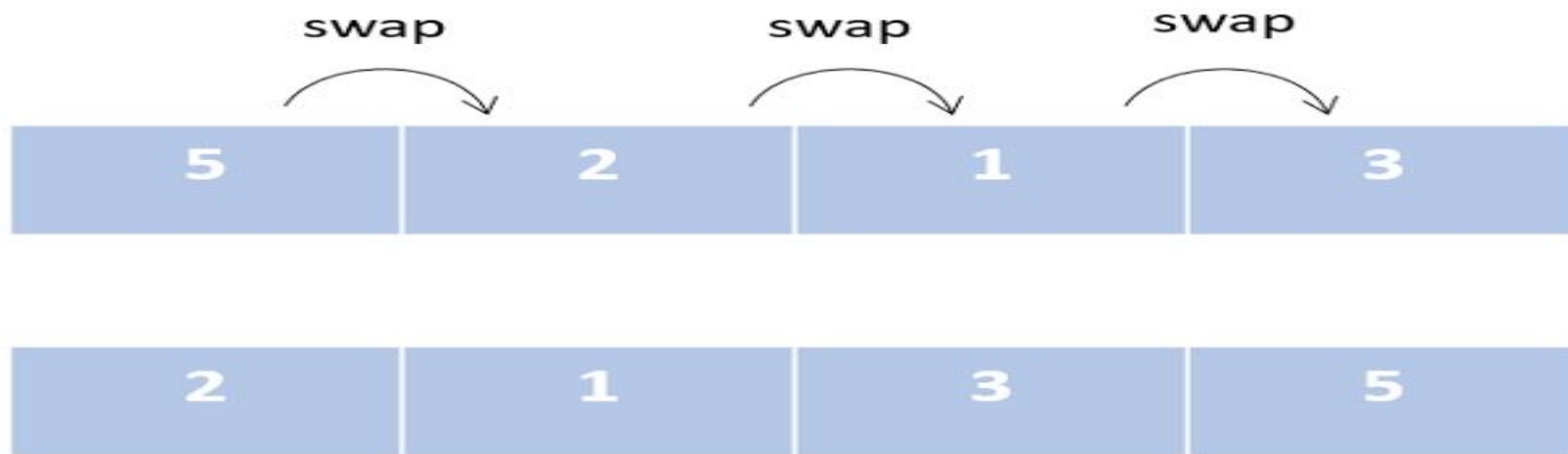
Сортировка пузырьком - это метод сортировки массивов и списков путем последовательного сравнения и обмена соседних элементов, если предшествующий оказывается больше последующего.

В процессе выполнения данного алгоритма элементы с большими значениями оказываются в конце списка, а элементы с меньшими значениями постепенно перемещаются по направлению к началу списка. Образно говоря, тяжелые элементы падают на дно, а легкие медленно всплывают подобно пузырькам воздуха.

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.

Количество итераций внутреннего цикла зависит от номера итерации внешнего цикла, так как конец списка уже отсортирован, и выполнять проход по этим элементам смысла нет.

Интерактив



Biggest element moves to right end

Еще один интерактив

исходный массив		обмен 2 и 3		обмен 2 и 7		обмен 2 и 5		нет обмена
1		1		1		1		1
5		5		5		2		2
7		7		2		5		5
3		2		7		7		7
2		3		3		3		3
первый проход циклом по массиву								

Двоичное(бинарное) дерево

Двоичное дерево в первую очередь дерево. В программировании – структура данных, которая имеет корень и дочерние узлы, без циклических связей. Если рассмотреть отдельно любой узел с дочерними элементами, то получится тоже дерево. Узел называется внутренним, если имеет хотя бы одно поддереву. Самые нижние элементы, которые не имеют дочерних элементов, называются листьями или листовыми узлами.

Первая особенность двоичного дерева, что любой узел не может иметь более двух детей. Их называют просто — левый и правый потомок, или левое и правое поддереву.

Вторая особенность двоичного дерева, и основное правило его построения, заключается в том что левый потомок меньше текущего узла, а правый потомок больше. Отношение больше/меньше имеет смысл для сравниваемых объектов, например числа, строки, если в дереве содержатся сложные объекты, то для них берётся какая-нибудь процедура сравнения, и она будет отрабатывать при всех операциях работы с деревом.

Интерактив

#1

Low	High	Mid
0	8	4

Search (45)

$$mid = \left[\frac{low + high}{2} \right]$$

0	1	2	3	4	5	6	7	8
6	12	17	23	38	45	77	84	90
↑ Low				↑ Mid				↑ High
38 < 45					→	Low = Mid + 1 = 5		

© w3resource.com

Домашнее Задание № 1

Доп Задача :

1. Перезаписать все две пройденные алгоритмы в ООП стиль (используя конструктор **init** и обращение к атрибуту через **self**)
2. К таким темам относятся **Сортировка Пузырьком** , **Быстрая сортировка**

Домашнее Задание № 2

Задача # легкий алгоритм

Смысл задачи: Дается число и если это число можно читать с конца до начало и выходит также как и при обратном счете значит это число относится к универсальным числам

1. Дано число 343 и нужно работать с ним (трехзначное число)
2. Если палиндром тогда return True
3. Все числа с минусовым значением заведомо считаются не универсальными такие как -343