Практическая работа №3

Реализация алгоритма сортировки методом пузырька:

Изображение выглядит как текст

Автоматически созданное описание

Реализация сортировки методом «пузырька» состоит из двух вложенных [циклов for](https://translate.google.com/translate?hl=ru&prev=_t&sl=en&tl=ru&u=https://realpython.com/python-for-loop/), в которых алгоритм выполняет n - 1*n*−1 сравнение, затем n - 2*n*−2 сравнения и так далее, пока не будет выполнено последнее сравнение в списке. В итоге получается (n - 1) + (n - 2) + (n - 3) + … + = n \times \frac{(n-1)}{2}(*n*−1)+(*n*−2)+(*n*−3)+…+=*n*×2(*n*−1)​ сравнения, которые также можно записать как \frac{1}{2}n^2 - \frac{1}{2}n21​*n*2−21​*n*, Ранее вы узнали, что «О»‑большое фокусируется на том, как увеличивается время выполнения в зависимости от размера входных данных. Это означает, что для того, чтобы превратить вышеприведенное уравнение в сложность алгоритма «О»‑большое, вам нужно удалить константы, потому что они не меняются в зависимости от размера ввода. Это упрощает запись до n^2 - n*n*2−*n*. Поскольку n^2*n*2 растет намного быстрее, чем n*n*, этот последний член также можно отбросить, оставляя пузырьковую сортировку со средней и наихудшей сложностью **O(n^2)*O*(*n*2)**. В тех случаях, когда алгоритм получает массив, который уже отсортирован — и при условии, что реализация включает в себя already sorted оптимизацию флага, описанную ранее, — сложность времени выполнения снизится до гораздо лучшего значения O(n)*O*(*n*), поскольку алгоритму не нужно будет сравнивать любой элемент более одного раза.

**Алгоритм поиска**

Линейный поиск — это один из самых простых и понятных алгоритмов поиска. Мы можем думать о нем как о расширенной версии нашей собственной реализации оператора in в Python.

Суть алгоритма заключается в том*,* чтобы перебрать массив и вернуть индекс первого вхождения элемента, когда он найден:

**def** LinearSearch(lys, element):

**for** i **in** range (len(lys)):

**if** lys[i] == element:

**return** i

**return** -1

Итак, если мы используем функцию для вычисления:

>>> print(LinearSearch([1,2,3,4,5,2,1], 2))

То получим следующий результат:

1

Это индекс первого вхождения искомого элемента, учитывая, что нумерация элементов в Python начинается с нуля.

Временная сложность линейного поиска равна O(n). Это означает, что время, необходимое для выполнения, увеличивается с увеличением количества элементов в нашем входном списке lys

**Контрольные вопросы:**

Алгоритм сортировки – это алгоритм упорядочивания элементов

Алгоритм поиска – это любой алгоритм, который решает задачи поиска (извлекает информацию, хранящуюся в некоторой структуре данных или вычисляемую в пространстве поиска проблемной области, либо с дискретными, либо с непрерывными значениями)