

Кружок по программированию на Python Занятие №15



Кружок по программированию на Python

Арифметические алгоритмы

- Сортировка пузырьком и почему так не надо сортировать
- Асимптотическая сложность алгоритма
- Сортировка слиянием
- Сортировка подсчетом
- Поразрядная сортировка
- Быстрая сортировка Хоара: Python





Введение

Сортировка пузырьком

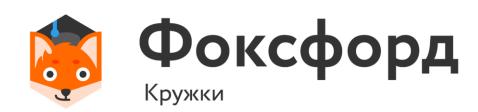
 На каждой итерации последовательно сравниваются соседние элементы, и, если порядок в паре неверный, элементы меняют местами. За каждый проход по массиву как минимум один элемент встает на свое место, поэтому необходимо не более n – 1 проходов, чтобы отсортировать массив.



Введение

Сортировка пузырьком

```
def bubble_sort(a):
  for i in range(len(a)-1):
    for j in range(len(a)-i-1):
        if a[j] > a[j+1]:
        a[j], a[j+1] = a[j+1], a[j]
```



Вычислительная сложность

Обозн.	Объяснение
f(n)=O(g(n))	f ограничена сверху функцией g (с точностью до постоянного множителя) асимптотически
$f(n) = \Omega(g(n))$	f ограничена снизу функцией g (с точностью до постоянного множителя) асимптотически
$f(n) = \Theta(g(n))$	f ограничена снизу и сверху функцией д асимптотически





Сортировка слиянием

- Если в рассматриваемом массиве один элемент, то он уже отсортирован алгоритм завершает работу;
- Иначе массив разбивается на две части, которые сортируются рекурсивно;
- После сортировки двух частей массива к ним применяется процедура слияния, которая по двум отсортированным частям получает отсортированный массив.





Быстрая сортировка Хоара

Quicksort

- из массива выбирается некоторый опорный элемент а[і];
- массив разделяется все элементы, меньшие, либо равные a[i] справа;
- теперь массив состоит из двух подмножеств, причем левое меньше, либо равно правого, далее рекурсивно.



Сортировка подсчётом

- Последовательно пройдём по массиву A и запишем в C[i] количество чисел, равных i.
- Теперь достаточно пройти по массиву С и для каждого числа number из диапазона допустимых значений последовательно записать в массив А число number C[number] раз.



Сортировка подсчётом

```
def SimpleCountingSort(A):
  scope = max(A) + 1
  C = [0] * scope
  for x in A:
    C[x] += 1
  A[:] = []
  for number in range(scope):
    A += [number] * C[number]
```



Поразрядная сортировка

- Создаем пустые списки, количество которых равно числу rang.
- Распределяем исходные числа по этим спискам в зависимости от величины младшего разряда (по возрастанию).
- Собираем числа в той последовательности, в которой они находятся после распределения по спискам.
- Повторяем пункты 2 и 3 для всех более старших разрядов поочередно.



Спасибо за внимание!