**Практическая работа 3**

**Оценка сложности алгоритмов сортировки. Оценка сложности алгоритма поиска.**

**Цель работы:** научиться разрабатывать алгоритмы сортировки и поиска, а также оценивать их сложность.

**Задание.**

1. Разработать алгоритм сортировки и оценить его сложность

Так как в задание не уточнялось как именно метод сортировки взять, мы выбрали метод «пузырька». Суть заключается в том, чтобы наша программа ставила числа по возрастанию.

Листинг

|  |
| --- |
| def bubble(nums):  swapped = True  while swapped:  swapped = False  for i in range(len(nums) - 1):  if nums[i] > nums[i + 1]:  nums[i], nums[i + 1] = nums[i + 1], nums[i]  nums = [3,4,2,5,1]  bubble(nums)  print(nums) |

Подсчитав всё, мы видим, что его сложность O(n^2). Это нам говорит что он достаточно простой, и времени тратим не очень много, ну смотря сколько чисел мы введем.

1. Разработать алгоритм поиска и оценить его сложность

В этом задании мы решили выбрать линейный поиск, так как он более понятен для первого решения и разбора.

Листинг

|  |
| --- |
| def linear(values, search\_for):  search\_at = 0  search\_res = False    while search\_at < len(values) and search\_res is False:  if values[search\_at] == search\_for:  search\_res = True  else:  search\_at = search\_at + 1  return search\_res  l = [22, 18, 17, 87, 93, 15, 76]  print(linear(l, 15)) |

Подсчитав всё, смело можно сказать, что сложность этого алгоритма равна О(n). Он довольно простой, и время зависит от количества вводимых чисел.

**Контрольные вопросы**

1. Что такое алгоритм сортировки?

Алгоритм сортировки – это алгоритм для упорядочения элементов в массиве

1. Что такое алгоритм поиска?

Алгоритм поиска – это любой алгоритм, который решает задача поиска, а именно, получить информацию, хранящуюся в некоторой структуре данных или вычисленную в пространстве поиска предметной области, либо с дискретными, либо с непрерывными значениями.

**Вывод:** мы научились разрабатывать алгоритмы сортировки и поиска, а также оценивать их сложность.