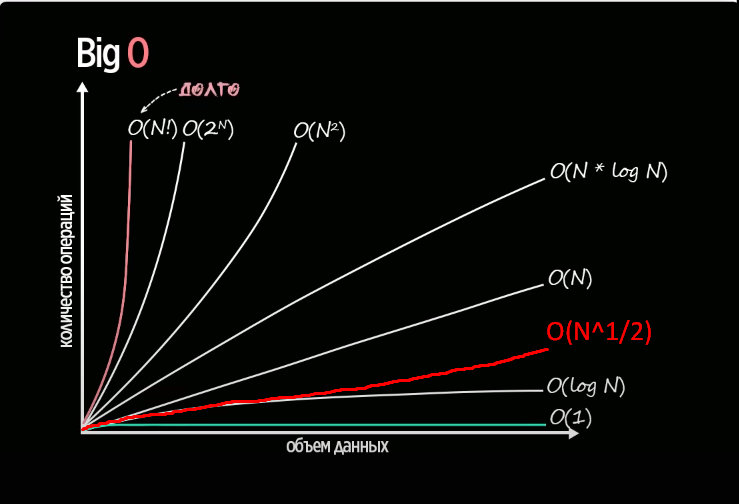
1. **Дополнительно**……………………………………………………...………….47
   1. Асимптотическая сложность алгоритмов…………………………………………50
2. **Дополнительно**
   1. **Асимптотическая сложность алгоритмов**

Для оценки сложности алгоритмов используется нотация «Big O», которая гласит: **от передаваемых параметров зависит количество операций, которые будут выполнены перед тем, как алгоритм завершится**.

|  |  |
| --- | --- |
| **Сложность** | **Описание** |
| факториальная **O(N!)** |  |
| экспоненциальная **O(2n)** |  |
| квадратичная **O(N2)** | **O(N\*N) ~ O(N2)**  **!** Степенью может быть **не только два**, но и другие числа.    **O(N\*N\*N) ~ O(N3)** |
| линейно-логарифмическая **O(N \* log N)** |  |
| линейная **O(N)** | Количество выполняемых операций алгоритмом прямо пропорционально величине входного параметра.    В данном случае сложность алгоритма оценивается как **O(2N) ~ O(N)**    **O(N+K)**    **O(N\*K)** |
| логарифмическая **O(log2 N)** | При длине массива в 16 элементов алгоритм сделает 5 шагов.  При длине массива в 8 элементов алгоритм сделает 4 шага.  Это выражается формулой **O(log2 N + 1)**, константа отбрасывается и получается **O(log2 N)** |
| сублинейная **()** | Количество операций, выполняемое алгоритмом, пропорционально квадратному корню величины входного параметра. На вход подаётся 100 -> 10 итераций, 16 -> 4 итерации. |
| константная **O(1)** | Независимо от входных параметров количество операций не изменяется (оно постоянно). |