**Classic bubble sort**

|  |  |
| --- | --- |
| N | Количество сравнений |
| 10 | 90 |
| 20 | 380 |
| 100 | 9900 |
| 1000 | 999000 |
| Теоретически | O(N2) |
| Практически | O(N2 - N) |

**Shaker sort**

|  |  |
| --- | --- |
| N | Количество сравнений |
| 10 | 45 |
| 20 | 190 |
| 100 | 4950 |
| 1000 | 499500 |
| Теоретически | O(N2) |
| Практически | O((N2 – N)/2) |

**CombSort** (то самое усовершенствование «пузырька»)

Основная идея в том, чтобы первоначально брать достаточно большое расстояние между сравниваемыми элементами и по мере упорядочивания массива сужать это расстояние вплоть до минимального. Первоначальный разрыв между сравниваемыми элементами лучше брать не абы какой, а с учётом специальной величины называемой **фактором уменьшения**. Опытным и теоретическим путём установлено оптимальное значение **фактора уменьшения**:

Сначала расстояние между элементами равно размеру массива разделённого на **фактор уменьшения**. Затем, пройдя массив с этим шагом, мы снова делим шаг на **фактор уменьшения** и проходим по списку вновь. Так продолжается до тех пор, пока разность индексов не достигнет единицы. В этом случае массив досортировывается обычным пузырьком.

|  |  |
| --- | --- |
| N | Количество сравнений |
| 10 | 36 |
| 20 | 121 |
| 100 | 1229 |
| 1000 | 22022 |

Подсчитать асимптотическую сложность алгоритма затрудняюсь.