**1.** Бульбашкове сортування:

Алгоритм:

* Починаємо з першого елемента масиву.
* Порівнюємо кожний елемент з наступним.
* Якщо елементи в неправильному порядку, міняємо їх місцями.
* Повторюємо цей процес до тих пір, поки не буде здійснено жодного обміну.

Асимптотика:

* У найкращому випадку: O(n), коли масив вже відсортований.
* У найгіршому випадку: O(n2), коли масив відсортований у зворотньому порядку.

Порівняння з сортуванням вставками:

* У найгіршому випадку обидва алгоритми мають асимптотику O(n2), але вставки зазвичай працюють краще на маленьких наборах даних через меншу константу у складності.

Порівняння з сортуванням злиттям:

* Бульбашкове сортування менш ефективне через свою квадратичну асимптотику, особливо на великих масивах даних, порівняно з сортуванням злиттям, яке має асимптотику O(n log(n)).

**2.** Сортування злиттям:

* Основна теорема рекурсії стверджує, що асимптотична складність сортування злиттям на масиві розміру n дорівнює: T(n) = 2T(n/2) + O(n), де T(n) - час сортування масиву розміру n.
* Застосовуючи основну теорему рекурсії, отримуємо, що асимптотична складність сортування злиттям - O(n log n).

**3.** Швидке сортування (алгоритм Квіка):

**Алгоритм:**

* Обираємо елемент, названий опорним.
* Розбиваємо масив на дві підмасиви: один містить елементи менші за опорний, інший - більші або рівні.
* Рекурсивно застосовуємо цей процес до кожного з підмасивів.
* Після рекурсивних викликів об'єднуємо підмасиви разом.

**Асимптотика:**

* У найкращому і середньому випадку: O(n log n), коли опорний елемент ділить масив на дві приблизно рівні частини.
* У найгіршому випадку: O(n^2), якщо на кожному кроці обирається найменший або найбільший елемент.

Швидке сортування зазвичай є більш ефективним за бульбашкове сортування та сортування злиттям на практиці через свою швидкодію на великих масивах даних. Але у разі незбалансованих даних або поганого вибору опорного елемента швидке сортування може показати гіршу продуктивність.