Para valores *crescentes*:

InsertionSort, pois é O(N) em comparação e em movimentação.

QuickSort, pois é O(N * logN) em comparação e em movimentação e ocorrerá, nessa implementação, o caso médio. Além disso, as constantes são menores do que outros algoritmos O(N * logN).

BinaryInsertionSort, pois é O(N * logN) em comparação, com constantes maiores do que as do QuickSort, mas é O(N) em movimentação.

BubbleSort, mas apenas em número de movimentação: O(1).

Para valores *decrescentes*:

Nesse caso, muitos algoritmos são O(N^2). Restam, assim, os O(N * logN). Dentre eles, o mais eficiente é o QuickSort.

Para valores *aleatórios*:

Como visto pelos gráficos, nesse caso também restam os O(N * logN). Dessa forma, prefere-se o QuickSort, que é o mais eficiente.