

① Bubble Sort 0

A complexidade do algoritmo é da ordem n^2 em todos os possíveis casos ($\Omega(n^2)$, $\Theta(n^2)$, $O(n^2)$), pois as iterações iniciadas nas linhas 9 e 10 serão sempre executadas em sua totalidade e como cada uma fará n iterações, logo teremos uma complexidade n^2 .

② Bubble Sort 1

O algoritmo apesar de apresentar uma melhoria em relação ao anterior, continua com complexidade da ordem n^2 para todos os casos. A melhoria consiste em ignorar a posição dos últimos elementos na segunda estrutura de repetição, pois como o algoritmo consiste em sempre tentar colocar um elemento por vez na sua posição ideal, então será apenas necessário percorrer $n-m$ posições na lista, onde m seria a quantidade de elementos já visitados na primeira iteração.

$$(n) \cdot (n-m) \Rightarrow \begin{aligned} &O(n^2) \\ &\Theta(n^2) \\ &\Omega(n^2) \end{aligned}$$

III BubbleSort 2

- Melhor Caso $\rightarrow \Omega(n)$

Como a lista está completamente ordenada, o algoritmo executará apenas uma única vez a primeira estrutura de repetição e após o fim da segunda estrutura de repetição o valor da variável "flag" continuará com o valor 0 (zero), pois não houve nenhuma troca entre os elementos, e realizará a quebra do primeiro laço. A segunda estrutura fará n passos.

- Pior Caso $\rightarrow O(n^2)$

Como a lista está ordenada de forma decrescente, o último elemento da lista estará no lugar correto, após $(n-1) \cdot (n-1)$ iterações, resultando em uma complexidade da ordem de n^2 .

- Caso médio $\Rightarrow O(n^2)$,

Caso limitado por n^2 . Caso o menor elemento estiver na última posição, haverá n^2 passos. Sempre serão executados mais de n passos, pois haverá pelo menos uma troca de posições entre os elementos.

IV BubbleSort 3

- Melhor Caso $\Rightarrow \Omega(n)$
- Caso Médio $\Rightarrow \Theta(n^2)$
- Pior Caso $\Rightarrow O(n^2)$

O algoritmo é uma mescla do bubbleSort 1 com o bubbleSort 2. A ideia é, além de passar a estratégia de ir diminuindo a quantidade do número de iterações da segunda estrutura de repetição, utilizar a flag para a quebra de iterações desnecessárias. Entretanto, em termos de notação assintótica, é semelhante ao BubbleSort 2.

V BubbleSort 4

- Melhor caso $\rightarrow \Omega(n)$
- Caso Médio $\rightarrow \Theta(n^2)$
- Pior Caso $\rightarrow O(n^2)$

Algoritmo semelhante ao BubbleSort 3, com a diferença que após a segunda iteração onde ao invés de apenas decrementar o valor da quantidade de iterações por 1, é realizada uma atribuição mais precisa na variável n .

⑥ BubbleSort 5

Melhor caso $\rightarrow \Omega(n)$

Caso Médio $\rightarrow \Theta(n^2)$

Pior Caso $\rightarrow O(n^2)$

Semelhante ao bubbleSort 4, mas não implementa a estratégia de decremento do número n de iterações.