

1. Baseado no array abaixo, desenvolva o seguinte:

[2, 7, 4, 9, 3]

Somente as alterações de troca no array:

- a. Passo a passo do Bubble Sort, Insertion Sort e Selection Sort
2. Baseado no array abaixo (em negrito) e o processo de ordenação, qual algoritmo produziria tais permutações?

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[1, 9, 8, 7, 6, 5, 4, 3, 2, 10]

[1, 2, 8, 7, 6, 5, 4, 3, 9, 10]

[1, 2, 3, 7, 6, 5, 4, 8, 9, 10]

[1, 2, 3, 4, 6, 5, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

3. Dado um array com elementos aleatórios, ao utilizar o algoritmo Selection Sort, seria executado em tempo linear, quadrático ou algo entre os dois anteriores? Justifique sua resposta.

4. Com base no método de ordenação a seguir, responda:

```
def odd_even_sort(a: np.array) -> None:
    for i in range(len(a)):
        if i % 2 != 0:
            for j in range(2, len(a), 2):
                if a[j] < a[j - 1]:
                    a[j], a[j - 1] = a[j - 1], a[j]
        else:
            for j in range(1, len(a), 2):
                if a[j] < a[j - 1]:
                    a[j], a[j - 1] = a[j - 1], a[j]
```

Complexidade de Espaço? (In-place ou Out-place)	
Baseado em comparação? (Sim/Não)	
Recursivo? (Sim/Não)	
Estável? (Sim/Não)	
Pior caso? $O(n)$ ou $O(n^2)$	