



Exercício por Temas

avale sua aprendizagem



ESTRUTURA DE DADOS EM PYTHON

Prezado (a) Aluno(a),

Você fará agora seu **EXERCÍCIO**! Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responde cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

ORDENAÇÃO

1. (MPE-BA/2023) Algoritmos de ordenação são responsáveis por ordenar elementos de uma estrutura de dados de forma completa ou parcial. Sobre a complexidade dos algoritmos de ordenação, assinale, a seguir, o algoritmo de ordenação que, no pior caso, tem complexidade igual a $O(n \log n)$.

- ☐ Bubble sort.
- ☐ Selection sort.
- ✓ ☒ Merge sort.
- ☐ Insertion sort.
- ☐ Quick sort.

Data Resp.: 19/08/2023 01:21:27

Explicação:

O Merge sort é um algoritmo de divisão e conquista que divide repetidamente a lista em sublistas menores, até que cada sublista contenha apenas um elemento. Em seguida, ele combina as sublistas em pares, ordenando-as ao mesmo tempo, e continua mesclando-as em sublistas maiores até que a lista esteja completamente ordenada. A complexidade do Merge sort no pior caso é $O(n \log n)$, onde "n" é o número de elementos a serem ordenados.

2. Avalie as afirmativas abaixo:

- 1 - O merge sort executa em $O(n \log n)$.
- 2 - O bucket sort executa em $O(n)$.

3 - Algoritmos que executam em uma complexidade abaixo de $O(n \log n)$ ordenam a sequência sem comparar os elementos desta sequência.

- ☐ Somente a 1 está correta.
- ☐ Somente a 1 e a 2 estão corretas.
- ✓ ☒ Todas estão corretas.
- ☐ Somente a 3 está correta.
- ☐ Somente a 2 e a 3 estão corretas.

Data Resp.: 19/08/2023 01:21:46

Explicação:

A resposta correta é: Todas estão corretas.

3. Algoritmos de ordenação baseados em comparação entre elementos da sequência tem complexidade computacional mínima de:

- ☐ $O(n)$
- ✗ ☒ $O(n^2)$
- ✓ ☐ $O(n \log n)$
- ☐ $O(n^3)$
- ☐ $O(\log n)$

Data Resp.: 19/08/2023 01:22:06

Explicação:

A resposta correta é: $O(n \log n)$

4. (UEAP/2014) Leia o seguinte trecho de código-fonte escrito em Java.

```
1. // laço com a quantidade de elementos do vetor (n = 10)
2. for(n = 1; n <= 10; n++) {
3.     for(i = 0; i <= 8; i++) {
4.         if(vet[i] > vet[i+1]) {
5.             aux = vet[i];
6.             vet[i] = vet[i+1];
7.             vet[i+1] = aux;
8.         }
9.     }
10. }
```

O algoritmo de ordenação implementado nesse trecho de código-fonte é o

- ☐ Quick Sort.
- ☐ May Sort.
- ☐ Heap Sort.
- ✓ ☒ Bubble Sort.
- ☐ Merge Sort.

Data Resp.: 19/08/2023 01:23:11

Explicação:

Bubble Sort indica um algoritmo de ordenação de elementos, que é um dos algoritmos mais simples e conhecidos para ordenar um conjunto de dados. O Bubble Sort funciona percorrendo a lista de elementos várias

vezes, comparando pares de elementos adjacentes e trocando-os se estiverem na ordem errada. O processo é repetido até que nenhuma troca seja necessária, o que significa que a lista está ordenada.

5. Comparando o Merge Sort com o Método da bolha podemos afirmar que:

- ☐ O merge sort sempre executará mais rápido que o bubble sort uma vez que sua complexidade é $O(n \log n)$ e a do bubble sort $O(n^2)$.
- ☐ Ambos têm complexidade comparável, assim, não é possível afirmar qual irá executar em melhor tempo.
- ☐ O merge sort, por ser instável, sempre executará em tempo superior ao bubble sort.
- ☐ O bubble sort sempre irá executar mais rápido que o merge sort por ter complexidade computacional inferior ao merge sort.
- ☒ O merge sort tem complexidade computacional inferior ao bubble sort, porém o merge sort sempre executa em um tempo proporcional a $n \log n$, enquanto o bubble sort, pode executar em tempo linear em algumas instâncias (melhores casos).

Data Resp.: 19/08/2023 01:24:09

Explicação:

A resposta correta é: O merge sort tem complexidade computacional inferior ao bubble sort, porém o merge sort sempre executa em um tempo proporcional a $n \log n$, enquanto o bubble sort, pode executar em tempo linear em algumas instâncias (melhores casos).

6. Todos os algoritmos de ordenação interna devem ter complexidade de espaço de:

- ☒ $O(n)$
- ☐ $O(1)$
- ☒ $O(n \log n)$
- ☐ $O(n^2)$
- ☐ $O(n^3)$

Data Resp.: 19/08/2023 01:23:25

Explicação:

A resposta correta é: $O(n)$

7. Os algoritmos de ordenação podem ser classificados de acordo com vários parâmetros diferentes. É um parâmetro comum utilizado para classificação:

- I. Caráter discursivo.
- II. Complexidade computacional.
- III. Complexidade de espaço.

Marque a alternativa correta.

- ☐ I, II e III.
- ☐ II, apenas.
- ☒ II e III, apenas.

☐ I e III, apenas.

☐ I, apenas.

Data Resp.: 19/08/2023 01:24:36

Explicação:

Os algoritmos de ordenação são frequentemente classificados de acordo com a complexidade computacional (tempo de execução) e a complexidade de espaço (quantidade de memória necessária). Esses dois parâmetros são importantes para avaliar o desempenho e a eficiência dos algoritmos de ordenação.

O parâmetro "caráter discursivo" mencionado na opção I não é um critério comum utilizado para classificar algoritmos de ordenação.

8. (FUMARC/2014 - Adaptada) Considere uma estrutura de dados do tipo vetor de tamanho 6 (seis) inicializado com as chaves abaixo:

	1	2	3	4	5	6
Vetor	79	82	68	69	78	65

Analise a ilustração dos passos intermediários de um método de ordenação das chaves do vetor:

	1	2	3	4	5	6
Vetor	79	82	68	69	78	65
Passo 1	65	82	68	69	78	79
Passo 2	65	68	82	69	78	79
Passo 3	65	68	69	82	78	79
Passo 4	65	68	69	78	82	79
Passo 5	65	68	69	78	79	82

O método de ordenação apresentado é:

- ☐ Shellsort.
- ☐ Ordenação por inserção.
- ☐ Mestrellasort.
- ✓ ☒ Ordenação por seleção.
- ☐ Quicksort.

Data Resp.: 19/08/2023 01:25:00

Explicação:

A ordenação por seleção é um algoritmo de ordenação que consiste em encontrar o menor valor no vetor e colocá-lo na primeira posição (ou o maior valor, dependendo da ordem desejada). Em seguida, encontra-se o segundo menor valor e o coloca na segunda posição, e assim por diante, repetindo o processo para os (n-1) elementos restantes, até os últimos dois elementos.

9. Sobre o método da bolha é correto afirmar que:

- ✓ ☒ O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada já estiver ordenada.
- ☐ O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada estiver ordenada em ordem reversa a desejada.
- ☐ O tempo de execução é definido pela complexidade computacional sempre, independentemente da instância apresentada.

- ☐ O algoritmo executa sempre no mesmo tempo para instâncias de mesmo tamanho n .
- ☐ A complexidade computacional deste algoritmo é $O(n \log n)$.

Data Resp.: 19/08/2023 01:25:30

Explicação:

A resposta correta é: O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada já estiver ordenada.

10. (CESGRANRIO/2014) Considere utilizar o algoritmo Bubble Sort para ordenar, em ordem crescente, a sequência de números

17, 43, 37, 31, 8, 77, 52, 25.

Se a sequência original for a iteração zero, qual será a sequência de números da segunda iteração?

- ☐ 17, 31, 37, 43, 8, 77, 52, 55
- ✔ ☒ 17, 31, 8, 37, 43, 25, 52, 77
- ☐ 17, 25, 37, 31, 8, 43, 52, 77
- ☐ 8, 17, 43, 37, 31, 77, 52, 25
- ☐ 17, 31, 8, 25, 37, 43, 77, 52

Data Resp.: 19/08/2023 01:26:06

Explicação:

Para resolver essa questão, vamos simular a execução do algoritmo Bubble Sort na sequência dada e acompanhar as trocas feitas na segunda iteração.

Sequência original: 17, 43, 37, 31, 8, 77, 52, 25

Primeira iteração:

17, 37, 31, 8, 43, 52, 25, 77 (comparação e troca: 43 e 37)

17, 31, 8, 37, 43, 25, 52, 77 (comparação e troca: 37 e 31)

17, 31, 8, 37, 25, 43, 52, 77 (comparação e troca: 43 e 25)

17, 31, 8, 37, 25, 43, 52, 77 (sem trocas)

Segunda iteração (a partir da sequência obtida na primeira iteração):

17, 31, 8, 37, 25, 43, 52, 77 (comparação e troca: 31 e 8)

17, 8, 31, 37, 25, 43, 52, 77 (comparação e troca: 37 e 8)

17, 8, 31, 25, 37, 43, 52, 77 (sem trocas)

17, 8, 31, 25, 37, 43, 52, 77 (sem trocas)

A sequência obtida na segunda iteração é: 17, 8, 31, 25, 37, 43, 52, 77.

 Não Respondida

 Não Gravada

 Gravada

Exercício por Temas iniciado em 19/08/2023 01:20:47.