
T160 - Raciocínio Lógico e Algorítmico

Prof. Bruno Lopes Alcantara Batista, Me

bruno.lopes@unifor.br

Algoritmos de Ordenação - Lista de Exercícios

Resolver o conjunto de exercícios relacionados a Recursão e Algoritmos de Ordenação conforme data especificada no Unifor Online. A resolução dos exercícios pode ser realizada em grupos de estudo e pode utilizar qualquer tipo de material de apoio (livros, sites, youtube, etc), além do uso de celular, computador, tablet, etc. **A entrega da atividade deve ser realizada individualmente, em formato ZIP, via Unifor Online** onde cada exercício deve ser colocado em uma pasta nomeada com o número do exercício.

Durante a realização dessa atividade é esperado que os alunos envolvidos desenvolvam as seguintes competências:

- Identificar problemas que tenham solução algorítmica;
- Resolver problemas com uso de ambientes de programação;
- Valorar a importância do pensamento computacional no cotidiano e suas aplicações em circunstâncias e domínios diversos;
- Gerir a sua própria aprendizagem e desenvolvimento, inclusive a gestão de tempo e competências organizacionais;
- Atuar colaborativamente em equipe de diferentes perfis e com os diversos níveis organizacionais.

Qualquer dúvida sobre a atividade ou sobre qualquer ponto que não esteja contemplado nesse documento, deve ser dirimida juntamente com o professor da disciplina.

Exercícios Propostos

1. Crie um vetor e o preencha com 100 números inteiros aleatórios, de 0 a 150. Escreva um algoritmo que busque um número qualquer nesse vetor e escreva se ele está ou não presente;
2. Implemente o algoritmo Bubblesort para ordenar um vetor de números inteiros em ordem decrescente;
3. Implemente o InsertionSort para ordenar um vetor de números inteiros em ordem crescente;
4. Implemente o Quicksort para ordenar um vetor de números inteiros em ordem crescente.
5. Crie dois vetores. Preencha o primeiro com 7 números inteiros aleatórios e o segundo com 15. Utilize um algoritmo de ordenação de sua preferência para ordená-los em ordem crescente. Preencha um terceiro vetor com todos os elementos do primeiro e do segundo, de forma que este terceiro vetor mantenha-se ordenado. Você não pode usar nenhum algoritmo de ordenação no terceiro vetor;
6. Implemente uma função recursiva que escreva os 20 primeiros números da sequência de fibonacci;
 - (a) Essa função leva um tempo acima do comum para calcular, por exemplo, os 100 primeiros números da sequência. Por que?
7. Escreva uma função que busque um caractere em uma string e retorne todas as posições em que ele se encontra. Exemplo, a frase "aprendemos mais quando temos que inventar." buscando pelo caractere 'a', teremos como saída do programa [0, 12, 18, 39].