



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

ESTRUTURA DE DADOS
TRABALHO FINAL

1. Escreva um programa em C para implementar o algoritmo *Insertion Sort*. Para testar o seu algoritmo na função *main*, crie um vetor de objetos implementado em uma TAD. Um exemplo seria criar um vetor de produtos, cada produto com um código e um valor, atribuir produtos para o seu vetor, e, depois, ordenar este vetor por meio do *Insertion Sort*. É preciso descrever o algoritmo, discorrendo sobre quem o criou, a sua lógica e a respectiva complexidade computacional. A descrição deve ser com as suas palavras e possuir ao menos 10 linhas.
2. Escreva um programa em C para implementar o algoritmo *Selection Sort*. Para testar o seu algoritmo na função *main*, crie um vetor de objetos implementado em uma TAD. Um exemplo seria criar um vetor de produtos, cada produto com um código e um valor, atribuir produtos para o seu vetor, e, depois, ordenar este vetor por meio do *Selection Sort*. É preciso descrever o algoritmo, discorrendo sobre quem o criou, a sua lógica e a respectiva complexidade computacional. A descrição deve ser com as suas palavras e possuir ao menos 10 linhas.
3. Escreva um programa em C para implementar o algoritmo *Bubble Sort*. Para testar o seu algoritmo na função *main*, crie um vetor de objetos implementado em uma TAD. Um exemplo seria criar um vetor de produtos, cada produto com um código e um valor, atribuir produtos para o seu vetor, e, depois, ordenar este vetor por meio do *Bubble Sort*. É preciso descrever o algoritmo, discorrendo sobre quem o criou, a sua lógica e a respectiva complexidade computacional. A descrição deve ser com as suas palavras e possuir ao menos 10 linhas.
4. Escreva um programa em C para implementar o algoritmo *Merge Sort*. Para testar o seu algoritmo na função *main*, crie um vetor de objetos implementado em uma TAD. Um exemplo seria criar um vetor de produtos, cada produto com um código e um valor, atribuir produtos para o seu vetor, e, depois, ordenar este vetor por meio do *Merge Sort*. É preciso descrever o algoritmo, discorrendo sobre quem o criou, a sua lógica e a respectiva complexidade computacional. A descrição deve ser com as suas palavras e possuir ao menos 10 linhas.
5. Escreva um programa em C para implementar o algoritmo *Quick Sort*. Para testar o seu algoritmo na função *main*, crie um vetor de objetos implementado em uma TAD. Um exemplo seria criar um vetor de produtos, cada produto com um código e um valor, atribuir produtos para o seu vetor, e, depois, ordenar este vetor por meio do *Quick Sort*. É preciso descrever o algoritmo, discorrendo sobre quem o criou, a sua lógica e a respectiva complexidade computacional. A descrição deve ser com as suas palavras e possuir ao menos 10 linhas.

6. Com base na complexidade computacional, qual você considerou como sendo o melhor algoritmo de ordenação abordado? Justifique sua resposta em no mínimo 8 linhas.