



Estrutura de Dados
Instituto Federal do Rio Grande do Norte
27 de julho de 2018

1 Ordenação

1. Implementar um programa que faça a ordenação de um vetor de 100 elementos gerados aleatoriamente via método Insert Sort.
2. Implementar um programa que calcule o tempo de ordenação de um vetor de 1000 elementos gerados aleatoriamente via método Insert Sort.
3. Implementar um programa que faça a ordenação de um vetor de 100 elementos gerados aleatoriamente via método Select Sort.
4. Implementar um programa que calcule o tempo de ordenação de um vetor de 1000 elementos gerados aleatoriamente via método Select Sort.
5. Implementar um programa que faça a ordenação de um vetor de 100 elementos gerados aleatoriamente via método Bubble Sort.
6. Implementar um programa que calcule o tempo de ordenação de um vetor de 1000 elementos gerados aleatoriamente via método Bubble Sort.
7. Implementar um programa que faça a ordenação de um vetor de 100 elementos gerados aleatoriamente via método Merge Sort.
8. Implementar um programa que calcule o tempo de ordenação de um vetor de 1000 elementos gerados aleatoriamente via método Merge Sort.
9. Implementar um programa que faça a ordenação de um vetor de 100 elementos gerados aleatoriamente via método Quick Sort.
10. Implementar um programa que calcule o tempo de ordenação de um vetor de 1000 elementos gerados aleatoriamente via método Quick Sort.
11. Implementar um programa que calcule o tempo de ordenação de um vetor de 2000 elementos gerados aleatoriamente via métodos Insert Sort, Select Sort, Bubble Sort, Merge Sort, Quick Sort, e exiba na tela o ranking dos tempo de cada método.

2 Lista

1. Usando listas ligadas, implementar uma agenda telefônica contendo nome e telefone, que permita as operações:
 - (a) Cadastro ilimitado de contatos
 - (b) Exclusão de contato a partir do nome
 - (c) Exclusão de um contato a partir de uma posição específica da lista;
 - (d) Recuperar o total de elemento da lista
 - (e) Listar todos os elementos da lista
 - (f) Procurar por um determinado contato na lista
2. Implementar uma lista duplamente ligada.
3. Implementar uma lista circular.

3 Pilha

1. Implementar uma pilha que possua os métodos de empilhar e desempilhar.
2. Escreva um programa para ler uma frase digitada via teclado, armazene cada letra em uma pilha, e como saída, exiba na tela a frase invertida.
3. Escreva um programa que use pilha para verificar se uma dada cadeia de caracteres é ou não palíndroma. Por exemplo: “subi no onibus” ou “amor a roma” é palíndroma.

4 Fila

1. Implementar uma fila que possua os métodos de enfileira e desenfileirar.
2. Implementar um programa que simule uma fila de espera de uma clínica médica para os pacientes serem atendidos em ordem de chegada.
3. Melhorar o programa da questão 2 para que a fila leve em consideração prioridade para os pacientes com mais de 65 anos de idade.

5 Árvore

1. Implementar um programa que monte uma árvore binária numérica.
2. Implementar um método de busca em profundidade dos elementos da árvore.
3. Implementar um método de busca em largura dos elementos da árvores.
4. Implementar um método para exclusão de elementos em uma árvore.
5. Implementar um método para fazer o balanceamento da árvore.
6. Implementar um programa que monte uma árvore de taxinomia de animais.
7. Implementar um programa que monte um árvore genealógica de uma família.