

4ª Lista de Exercícios

Estrutura de Dados

Prof. Hamilton José Brumatto

Pilhas

1. Construa uma rotina que execute uma operação de *troca* em uma pilha. Esta operação deve trocar de posição o primeiro e segundo elemento do topo da pilha. Atenção aos casos nos quais a operação não pode ser aplicada.
2. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma:
x C y
onde x é uma string consistindo nas letras 'A' e 'B', e y é o seu inverso. Por exemplo, a string ABBCBBA é válida.
3. Altere a implementação apresentada de pilha para que os objetos inseridos na pilha possam ser, por exemplo, números racionais, ou complexo.
4. Crie uma implementação de pilhas usando orientação a objetos. Esta pilha poderá utilizar-se de Templates para inserir qualquer tipo de objeto.
5. Considere o seguinte algoritmo de ordenação por troca: MERGE_SORT.
Neste algoritmo, uma sequência é dividida em duas e, recursivamente, cada metade estará ordenada. O passo do algoritmo é intercalar ambas metades em uma única sequência ordenada. A recursividade termina quando o tamanho da sequência é um, $p \geq r$.

Entrada: Uma sequência de números A e os índices de início da sequência a ser ordenada: p e r

Saída: A mesma sequência, no intervalo $[p, r]$, de números, ordenada

Algoritmo MERGE_SORT(A, p, r)

se $p < r$ **então**

$q \leftarrow (p + r) / 2$

 MERGE_SORT(A, p, q)

 MERGE_SORT($A, q + 1, r$)

 INTERCALA(A, p, q, r)

A função INTERCALA é realizada da seguinte forma: a primeira metade na sequência e a segunda metade copiada na sequência inversa. Em seguida copia-se das pontas do menor para o maior em uma sequência única

Entrada: Uma sequência onde no intervalo $[p, r]$ a sequência está ordenada no subarranjo $[p, q]$ e no subarranjo $[q+1, r]$.

Saída: A mesma sequência, ordenada no intervalo $[p, r]$.

Algoritmo INTERCALA(A, p, q, r)

para $i \leftarrow p$ **até** q **faça**

$B[i] \leftarrow A[i]$

para $j \leftarrow q + 1$ **até** r **faça**

$B[r + q + 1 - j] \leftarrow A[j]$

$i \leftarrow p$

$j \leftarrow r$

para $k \leftarrow p$ **até** r **faça**

se $B[i] \leq B[j]$ **então**

$A[k] \leftarrow B[i]$

$i \leftarrow i + 1$

senão

$A[k] \leftarrow B[j]$

$j \leftarrow j - 1$

Pede-se: Transforme o algoritmo MergeSort(A, p, r) em uma versão não recursiva utilizando pilha para simular a recursão. Observe que os parâmetros empilhados são: p e r .