

Algoritmos e Estrutura de Dados I

Fabício Olivetti de França e Paulo Henrique Pisani

01 de março de 2019

Lista de Exercícios 1

- 1) Utilizando as equações do slide 30 e 31 da aula 02, defina analiticamente o valor de p_{n0} .
- 2) Utilizando informação do slide 36 da aula 02, responda qual o mínimo, máximo e média de execuções do passo L4 do algoritmo `maximo` para $n = 1000$.
- 3) Multiplique $(\ln n + O(1/n))$ por $(n + O(\sqrt{n}))$ e expresse a resposta em notação-O. Use a tabela de propriedades do slide 51 da aula 02.
- 4) Crie uma estrutura de dados para representar números complexos. Implemente as operações de soma, multiplicação e negação.
- 5) Um deque com entrada-restrita é aquele que os itens podem ser inseridos em um dos lados e removido de qualquer um dos dois lados. Essa estrutura pode ser utilizada como uma pilha ou fila, se removermos os itens sempre de apenas um dos lados. Um deque com restrição de saída também possui tal propriedade?
- 6) Imagine quatro trens posicionados na entrada da figura do slide 24 da aula 03, eles são numerados de 1 a 4 da esquerda para direita. Suponha a seguinte sequência de movimentos dos trens:
 - Trem 1 para a pilha
 - Trem 2 para a pilha
 - Trem 2 para a saída
 - Trem 3 para a pilha
 - Trem 4 para a pilha
 - Trem 4 para a saída
 - Trem 3 para a saída
 - Trem 1 para a saída

Com essa sequência a ordem dos trens foram alteradas de 1234 para 2431. Se tivermos 6 carros na sequência 123456, é possível permutá-los na ordem 325641? E na ordem 154623? Mostre a sequência de operações.

- 7) Em uma pilha de tamanho 5, quais dessas seqüências de operações causa overflow ou underflow?
- Push, Push, Pop, Push, Push, Push, Pop, Push, Push, Pop, Push, Push
 - Push, Push, Push, Push, Pop, Pop, Pop, Push, Pop, Pop, Pop
 - Push, Pop, Push, Pop, Push, Push, Pop, Pop, Push, Pop, Pop, Push
 - Push, Pop, Push, Push, Pop, Push, Push, Push, Pop
 - Push, Push, Push, Pop, Pop, Push, Pop, Pop, Push, Pop
- 8) Implemente uma pilha de caracteres. Utilize essa pilha para verificar se uma expressão como $\{x + [2 * (y + z)]\}$ abre e fecha todos os parênteses, colchetes e chaves na ordem correta. Para isso, toda vez que receber um dos caracteres $\{(, [, \{$, eles devem ser empilhados, toda vez que ler um caractere $\})]$ devemos desempilhar e verificar se estamos fechando corretamente. Quais condições devem ser verificadas?
- 9) Simule o procedimento acima para as expressões:
- $(A + B)$
 - $\{[A + B] - [(C - D)]\}$
 - $(A + B) - \{C + D\} - [F + G]$
 - $((H) * \{([J + K])\})$
 - $((((A))))$
- 10) Implemente uma pilha utilizando apenas um `int s[SIZE]` em que `s[0]` é utilizada para apontar para o topo da pilha e `s[1]` a `s[SIZE-1]` contém os elementos da pilha. Crie as funções `push` e `pop` para essa estrutura.
- 11) Como você implementaria uma fila de pilhas? E uma pilha de filas? E uma fila de filas? Escreva rotinas para implementar as operações corretas para cada uma destas estruturas.
- 12) Faça um programa que receba duas strings e compare se uma é sufixo da outra. Utilizamos pilha ou fila?
- 13) Escreva as seguintes funções para uma lista simplesmente ligada:
- Inserir um elemento no final da lista: modifique a função que escrevemos em aula de modo que essa operação seja $O(1)$, ao invés de $O(n)$
 - Concatenar duas listas
 - Inverter uma lista de modo que o último elemento se torne o primeiro e assim por diante (não crie outra lista: a inversão deve ser realizada na própria lista)
 - Remover o último elemento de uma lista
 - Remover o n ésimo elemento de uma lista
 - Liberar a memória de todos os nós da lista
 - Combinar duas listas ordenadas em uma única lista ordenada
 - Formar uma lista contendo a união dos elementos de duas listas ordenadas
 - Formar uma lista como a interseção dos elementos de duas listas ordenadas

- j. Eliminar cada segundo elemento de uma lista
 - k. Retornar a soma dos elementos de uma lista de inteiros
 - l. Deslocar um ponteiro N nós para frente
 - m. Em uma lista ordenada, remova os elementos duplicados
 - n. Em uma lista com números inteiros, remova todos os valores negativos
 - o. Em uma lista com números inteiros, duplique os nós com valores pares
 - p. Quebre uma lista ligada na metade. Se a quantidade elementos da lista original for ímpar, a primeira metade ficará um elemento a mais.
 - q. Implemente a função `double get_valor(int indice, linked_node *inicio)`, que retorna o valor armazenado no índice indicado de uma lista ligada de `double`. Caso o índice seja negativo, comece a conta a partir do último elemento da lista. Por exemplo, o `get_valor(-2, inicio)` retorna o penúltimo elemento da lista.
- 14) Faça versões das funções acima que possam tratar listas duplamente ligadas
- 15) Implemente pilha e fila com lista simplesmente ligada