Lista 1 - Pilha

ESTRUTURA DE DADOS I – Pedro Nuno Moura

Monitor: Celio Ferreira Camara Junior

1) Escreva uma função para determinar se uma cadeia de caracteres (string) é da forma: x C y onde x e y são cadeias de caracteres compostas por letras ‘A’ e/ou ‘B’, e y é o inverso de x. Isto é, se x = “ABABBA”, y deve equivaler a “ABBABA”. Em cada ponto, você só poderá ler o próximo caractere da cadeia.

2) Escreva uma função para determinar se uma cadeia de caracteres (string) é da forma: a D b D c D ... D z onde cada cadeia de caracteres, a, b, ..., z, é da forma do exercício descrito acima. Portanto, uma cadeia de caracteres estará no formato correto se consistir em qualquer número de cadeias deste tipo (x C y), separadas pelo caractere ‘D’. Em cada ponto, você só poderá ler o próximo caractere da cadeia (é mandatório o uso de pilha).

3) Desenvolva um método para manter duas pilhas dentro de um único vetor (*array*) de modo que nenhuma das pilhas incorra em estouro até que toda a memória seja usada, e toda uma pilha nunca seja deslocada para outro local dentro do vetor.

4) Utilizando as operações de manipulação de pilhas vistas em aula, assim como o código de PilhaGenerica visto, use uma pilha auxiliar e uma variável do tipo T, para desenvolver um procedimento que remova um dado objeto do tipo T de uma posição qualquer de uma pilha. Para saber se dois objetos do tipo T são iguais, você deve usar o método *equals* (ou *compareTo*). Note que você não pode acessar diretamente a estrutura interna da pilha (atributos), devendo usar apenas as operações (métodos) de manipulação.

5) Escreva um programa que leia uma sequência de caracteres e determine se os parênteses, colchetes e chaves estão balanceados. Se a sequência não possuir esses caracteres ele deve ser considerado balanceado. Exemplo:

“{ab}[cde]” - Balanceado
“{ab[cd]efg}” - Balanceado
“[abcde{efg]}” - Não balanceado

6) Elabore um método que retorne as letras invertidas das palavras de uma frase recebida por parâmetro, preservando a ordem das palavras na frase. Por exemplo “a maçã está podre”, deve ter como saída: “a ãçam átse erdop”. As operações básicas de uma pilha, *push* e *pop*, devem ser usadas.

7) Em sala de aula, aprendemos a avaliar uma *expressão aritmética* que estivesse em notação pós-fixada, que não faz uso de parênteses por não possuir ambiguidade na sua avaliação. Neste exercício, você deve utilizar o conceito de Pilha para realizar a avaliação de expressões aritméticas em notação *infixa*, isto é, aquela que aprendemos e utilizamos ao longo do Ensino Fundamental e do Ensino Médio. Nesse contexto, vamos usar a seguinte

definição recursiva: uma expressão aritmética é um número, ou um parêntese esquerdo seguido de uma expressão aritmética seguida por um operador seguido por outra expressão aritmética seguida de um parêntese direito. Por simplicidade, essa definição assume que a expressão está *completamente parentizada*, especificando precisamente quais operadores devem ser aplicados a quais operandos e removendo possíveis ambiguidades na avaliação. Por exemplo, a expressão (1 + ((2 + 3) * (4 * 5))) segue essa definição.

Você deve então implementar um método que realize a avaliação de tais expressões representadas em uma String usando o conceito de Pilha. Por fim, cabe citar que o código que voces vão desenvolver corresponde a um exemplo simples de um *interpretador*: um programa que interpreta uma computação especificada por uma string e realiza tal computação para chegar ao resultado.

8) Dada uma sequência de 1 a N armazenada em um *array*, são formadas todas as subsequências (*subarrays*) possíveis a partir da sequência original. Para todas essas subsequências geradas, encontre a quantidade de pares únicos (a, b), em que ‘a’ é diferente de ‘b’ e ‘a’ é máximo (maior número) e ‘b’ é o segundo máximo da subsequência. Por exemplo, em uma sequência de 1 até 5, podem ser formadas as seguintes 15 subsequências:

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
2
2 3
2 3 4
2 3 4 5
3
3 4
3 4 5
4
4 5
5

Nessas 15 subsequências, existem 4 pares únicos que satisfazem aos critérios definidos: (2,1), (3,2), (4,3) e (5,4).