

MAC 323 - Algoritmos e Estruturas de Dados II**Primeiro semestre de 2022**

Lista de Exercícios para auto-avaliação

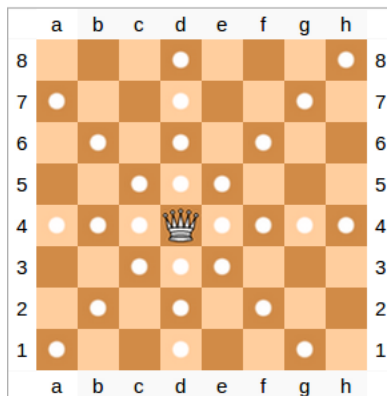
Não é necessário entregar nenhum exercício. A ideia é que cada um possa verificar se está acompanhando a disciplina

Pilhas e Filas

1. As operações de colocar e tirar os n vagões do estacionamento podem ser codificadas concisamente usando a letra E para “empilhar” (ou colocar um vagão no estacionamento) e D para “desempilhar” (ou tirar um vagão). Chamamos uma sequência de E’s e D’s de *admissível* se contém n E’s, n D’s e as operações codificadas podem ser realizadas. Por exemplo, a sequência EDEEEEDDEEEEDDDDD é admissível, enquanto a sequência EDDEEEEDD não é admissível. Formule uma regra que permita diferenciar as sequências admissíveis das que não são.
2. A partir do exercício acima, derive uma fórmula simples que, dado n , você possa calcular o número de sequências de n E’s e n D’s válidas.
3. Mostre que é possível obter uma permutação p_1, p_2, \dots, p_n a partir de $1, 2, \dots, n$ usando uma pilha se e somente se não existirem índices i, j e k tais que $i < j < k$ e $p_k < p_i < p_j$.
4. Faça um algoritmo que dado uma permutação p_1, p_2, \dots, p_n de $1, 2, \dots, n$ verifique se a permutação pode ser obtida utilizando uma pilha e caso seja, retorne a sequência de operações (E’s e D’s) que devem ser realizadas.
5. Passe a expressão aritmética abaixo para a notação posfixa, indicando para cada caractere lido o conteúdo da pilha de operadores.

$$A + B * (C + D) / E - B - D$$

6. Resolva a expressão posfixa do exercício acima utilizando como valores: $(A, B, C, D, E) \rightarrow (7, 10, 3, 9, 4)$. Mostre o conteúdo da estrutura de dados que você utilizar após cada passo.
7. O **problema das 8 rainhas** consiste no seguinte: dado um n , determinar se existe uma maneira de colocar n rainhas num tabuleiro de xadrez $n \times n$ sem que nenhuma delas ataque a outra. Escreva um algoritmo que resolva o problema das 8 rainhas. Caso exista uma maneira de colocar n rainhas, seu algoritmo deve imprimir as posições das n rainhas para uma destas maneiras válidas de colocá-las. (No xadrez, uma rainha ataca qualquer peça que esteja na mesma linha, coluna ou diagonal que ela. Veja figura abaixo).



8. Um famoso *site* de relacionamentos está desenvolvendo uma nova *feature*. Eles informam a você uma pessoa que é bastante distante do seu círculo de amigos. Para isso desenvolveram a seguinte métrica, chamada *grau de separação*:
Eles têm a lista de todos os seus amigos. O valor do seu grau de separação para cada amigo é 1. Para cada amigo dos seus amigos, seu grau de separação é 2. E assim por diante.
Mas você não quer esperar, e resolveu achar essas pessoas “distantes” por conta própria. Faça uma função que receba uma lista de todas as pessoas do *site* de relacionamentos e a lista de amigos de cada uma dessas pessoas e que devolva uma das pessoas com maior grau de separação em relação a você mesmo.
9. Digamos que nosso alfabeto é formado pelas letras **a**, **b** e **c**. Considere o seguinte conjunto de cadeias de caracteres sobre nosso alfabeto:

c, aca, bcb, abcba, bacab, aacaa, bbcbb, ...

Qualquer cadeia deste conjunto tem a forma WcM , onde W é uma sequência de letras que só contém **a** e **b** e M é o inverso de W , ou seja, M é W lido de trás para frente. Escreva um algoritmo que determina se uma cadeia X pertence ou não ao nosso conjunto, ou seja, determina se X é da forma WcM .

Apontadores, listas ligadas e árvores

1. Crie uma função iterativa que receba um ponteiro para o início de uma lista, inverta-a, e devolva o ponteiro para o novo início da lista. Faça também uma versão recursiva. Atenção, você não pode modificar o valor dos nós, apenas os ponteiros.
2. Crie uma função que receba o ponteiro para o início de uma lista e ordene-a em ordem crescente do conteúdo dos nós. Atenção, você não pode modificar o valor dos nós, apenas os ponteiros.
3. Faça uma função que receba ponteiros para duas listas de vértices de uma árvore binária visitados em “in-ordem” e “pre-ordem” e constrói a árvore binária correspondente.
4. Escreva uma função que receba uma lista encadeada e devolva um ponteiro para o nó que esteja o mais próximo possível do meio da lista. Faça isso sem contar explicitamente o número de nós da lista.

Árvores de busca binária balanceadas

1. Mostre o formato de uma árvore rubro-negra inicialmente vazia após a inserção de cada elemento abaixo:

15 10 7 18 25 11 23 29 33

2. Desenhe uma árvore rubro-negra com $n = 20$ vértices de altura máxima.
3. Considere uma árvore 2-3.
 - a. Se a árvore tem $n = 30000$ elementos, qual é a altura mínima e máxima que ela poderá ter?
 - b. Descreva a estrutura de dados para representar uma árvore 2-3.
4. a. Simule o algoritmo de inserção em árvores 2-3 para a montagem da árvore correspondente à sequência abaixo, mostrando detalhadamente os rearranjos que ocorrerem.

43 12 10 4 3 7 9 33 22 19 14 13 17 39

- b. Remova da árvore obtida no item acima, sucessivamente os elementos 4, 3, 7 e 9. (simule detalhadamente).