

1) Implementar um TAD lista e suas operações básicas. Tratar os casos possíveis de inserção e remoção.

2) Escrever uma função recursiva para imprimir um vetor.

3) Escrever uma função recursiva para imprimir um vetor de trás para a frente.

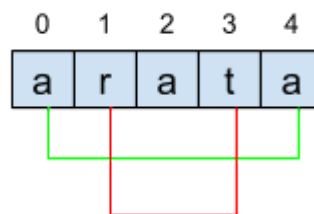
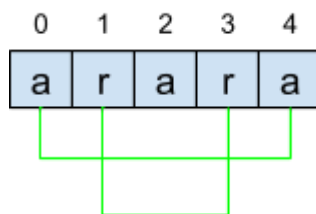
4) Escrever uma função recursiva para verificar se uma string é palíndromo. Ideia: verificar se:

1º == último

2º == penúltimo

3º == antepenúltimo

...



5) Escrever uma função recursiva que retorne a quantidade de dígitos em um número inteiro.

6) Escrever uma função recursiva para verificar se um número é primo ou não.

7) Qual dos seguintes algoritmos têm a menor complexidade de tempo no pior caso:

- a) Bolha.
- b) Quick sort.
- c) Merge sort.
- d) Inserção.

8) Qual dos seguintes algoritmos de ordenação será mais eficiente se aplicado a um vetor ordenado ou quase ordenado:

- a) Quick sort.
- b) Merge sort.
- c) Seleção.
- d) Inserção.

9) Qual dos seguintes algoritmos de ordenação será mais eficiente se aplicado a um vetor onde todos os elementos são iguais:

- a) Merge sort.
- b) Quick sort.
- c) Seleção.
- d) Inserção.

10) Uma versão do algoritmo quick sort sempre seleciona o $n/4$ -ésimo menor elemento do vetor como pivô. Considere que o custo para selecionar este pivô é $O(n)$. Qual seria a complexidade do quick sort nessa situação?

11) Elaborar uma versão da busca binária para um vetor decrescente.