

Lista de Exercícios – 3
Disciplina Algoritmo Estrutura de Dados - I
Prof. Wagner dos Santos C. de Jesus

Pedro Henrique Domiciano Alves

01910210

Curso Engenharia da Computação

- 1) Efetue a leitura dos conceitos do slide-6 do material – 1 e comente.

TAD's Basicamente permite que sejam manipulados os varios tipos de dados que estão dentro de uma lista

- 2) Escreva três tipos principais de estrutura de dados.

Array, Lista e Pilha

- 3) Comente as vantagens e desvantagens de uma lista usando Arranjos.

Vantagens: economiza mais memoria

Desvantagens: para efetuar uma inclusão ou remoção de um elemento, é necessario deslocar outros itens.

- 4) Qual a função do Garbage Collection?

É responsavel por liberar a memoria que está sendo usada mais no programa

- 5) Sobre alocação dinâmica, é correto, afirmar que não existe de fato esse tipo de recurso, por quê?

A alocação dinâmica permite que a gente reserve memória durante a execução do programa, mas ela não é infinita. A alocação dinâmica ainda depende da quantidade de memória disponível no sistema, ou seja, se a memória acabar, o programa pode falhar ao tentar alocar mais espaço.

- 6) Explique o processo de fragmentação em memória.

A fragmentação acontece quando a memória livre fica espalhada por diferentes partes da memória física. Isso pode causar problemas quando tentamos alocar um grande bloco de memória, já que, apesar de haver espaço livre suficiente no total, ele não está contíguo.

- 7) Determine o que vem a ser uma variável?

É um espaço na memoria que é possivel definir um valor

8) O que vem a ser uma lista linear?

É uma lista que segue uma ordem, ou seja, sempre terá um item antes e depois de um valor (com exceção do primeiro valor e do último)

9) Explique quais são as propriedades de uma lista linear.

O primeiro elemento não tem antecessor.

O último elemento não tem sucessor.

Os elementos intermediários têm um antecessor e um sucessor.

O número de elementos na lista pode variar.

10) Comente porque o algoritmo apresentado é ineficiente, proponha alguma técnica, para resolver esse problema.

```
leia n
Se n ≠ 0 então
    String:Lista[n]
    Para k de 0 até n
        leia elem
        lista[k] <- elem
    Fim_para
Senão
    escreva "Lista Vazia"
Fim_se
```

O algoritmo apresentado é ineficiente porque, em arranjos, inserir ou remover um elemento no meio da lista pode exigir o deslocamento de vários outros elementos, o que é custoso. Uma solução mais eficiente seria usar uma **lista encadeada**, onde cada elemento aponta para o próximo, o que permite inserções e remoções rápidas, sem a necessidade de deslocar elementos.