

**Estruturas de Dados Avançada**  
**Lista de exercícios 1 - Parte I**

1. Para o algoritmo  $\text{Alg1}(A)$  (descrito abaixo) faça:
  - (a) encontre a função geral de tempo do algoritmo  $\text{Alg1}$ .
  - (b) a partir da função geral, calcule a complexidade de tempo de melhor caso e em seguida mostre qual conjunto, em notação  $\Theta$ , a função de melhor caso pertence. Informe também qual o tipo de vetor que leva para o tempo de melhor caso.
  - (c) a partir da função geral, calcule a complexidade de tempo de pior caso e em seguida mostre qual conjunto, em notação  $\Theta$ , a função de pior caso pertence. Informe também qual o tipo de vetor que leva para o tempo de pior caso.

---

**Algoritmo 1:**  $\text{Alg1}(A)$

---

```
1 para  $i \leftarrow 1$  até  $n - 1$  faça
2    $j \leftarrow i$ 
3   enquanto  $(A[i] \% 2 == 0) \ \&\& \ (j > 0)$  faça
4     escreva ("*")
5      $j \leftarrow j - 1$ 
```

---

2. Prove as seguintes afirmações sobre notação assintótica:
  - (a)  $n^3/100 - 25n^2 - 100n + 7$  é  $\Theta(n^3)$
  - (b)  $77n^3 - 13n^2 + 29n - 5$  é  $\Omega(n^3)$
  - (c)  $34n \log_7 n^2 + 13n$  é  $\Omega(n)$  e  $O(n^2)$
  - (d) sendo  $f(n)$  e  $g(n)$  duas funções, se  $f(n) = O(g(n))$  e  $g(n) = O(f(n))$ , então  $f(n) = \Theta(g(n))$ .
  - (e)  $n^3 \neq O(n^2)$  (entenda  $n^3 \notin O(n^2)$ ).
  - (f)  $\log_a n = \Theta(\log_b n)$  para quaisquer  $a, b > 1$ .
  - (g)  $\log n! = O(n \log n)$ .
3. Verifique se estas sequências correspondem ou não a um heap:
  - (a) 33 32 28 31 26 29 25 30 27
  - (b) 33 32 28 31 29 26 25 30 27
4. Faça os algoritmos  $\text{subir}(i)$  e  $\text{descer}(i, n)$ , para um heap, sem utilizar recursividade. Qual a complexidade de tempo de cada um dos seus algoritmos?
5. Seja a sequência 18 25 41 34 14 10 52 50 48. Mostre passo a passo do algoritmo para construir o heap com a sequência dada.
6. Prove ou dê um contraexemplo:

- (a) Seja a sequência  $S = (s_1, s_2, \dots, s_n)$  um heap. Se existem  $s_i$  e  $s_j$  tais que  $s_i < s_j$  e  $i < j$ , então ao troca  $s_i$  por  $s_j$  na sequência  $S$  ainda continua um heap.
- (b) Análogo ao item (a) com  $s_i > s_j$  e  $i < j$ .
- 7.** Altere o algoritmo HEAPSORT para trabalhar com Heaps mínimos, ao invés de Heaps máximos. Argumente porque é melhor trabalhar com Heaps máximos ao invés de Heaps mínimos.
- 8.** Seja a sequência  $S = (0, 7, 14, 21, \dots, 7 \times (k - 1))$  dos  $k$  primeiros múltiplos de 7. Para uma tabela de dispersão, quantas colisões temos para a função de dispersão:
- (a)  $x \bmod 7$
- (b)  $x \bmod 14$
- (c)  $x \bmod 5$
- 9.** Faça os algoritmos de busca, inserção e remoção em uma tabela de dispersão com tratamento de colisões pelo método de encadeamento exterior.
- 10.** Seja  $S = \{1, 2, 3, 4, 5, 6, 7\}$ . Desenhar uma árvore binária de busca com os 7 elementos de  $S$  em que a altura seja máxima. Desenhar outra com altura mínima. Quantas árvores distintas existem em cada caso? E quando  $|S| = k$  quantas árvores distintas existem em cada caso?
- 11.** Mostre o passo a passo para obter uma AVL com as inserções na seguinte sequência de chaves: 8 5 7 4 12 14 10 9 3 6 11 13. Mostre o que ocorre ao excluir 6.
- 12.** Defina recursivamente o número de árvores AVL distintas com tamanho  $h$ .
- 13. (Poscomp)** Considere as seguintes estruturas de dados:
- (I) Tabela hash
- (II) Fila
- (III) Árvore de binária de busca
- (IV) Pilha
- Qual ou quais das estruturas acima requer mais do que tempo médio constante para inserção de um elemento?
- (a) Somente(I)
- (b) Somente(II)
- (c) Somente(III)
- (d) Somente(IV)
- (e) Todas
- 14. (Enade)** Uma sequência desordenada de números armazenada em um vetor é inserida em uma árvore AVL. Após a inserção nesta árvore, é feito um percurso em ordem simétrica (em ordem) e o valor de cada nó visitado é inserido em uma pilha. Depois de todos os nós serem visitados, todos os números são retirados da pilha e apresentados na tela. A lista de números apresentada na tela está:
- ( ) ordenada ascendentemente de acordo com os números.
- ( ) ordenada descendentemente de acordo com os números.
- ( ) na mesma ordem do vetor original.
- ( ) na ordem inversa do vetor original.