

# Algoritmos e Estrutura de Dados

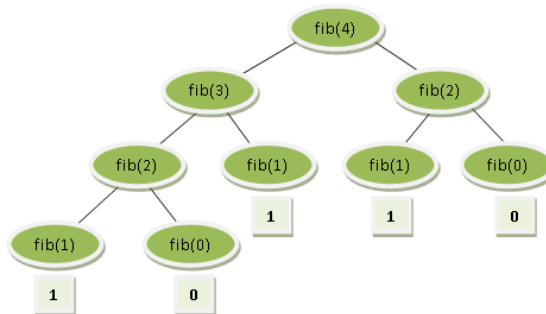
## Lista de Questões para a P1 – Pedro Achcar

Fazer a lista usando as estruturas de dados aprendidas até então (fila, pilha, deque e lista) e dizer qual é o big  $\Omega$  e o big  $O$  do problema e da sua solução, respectivamente.

- 1) Implemente sua versão das estruturas de dados básicas (não utilize uma lista já pronta, faça com nós soltos, usando classes no Python por exemplo, como visto na aula/monitoria):
  - a. Pilha
  - b. Fila
  - c. Deque
  - d. Lista Encadeada
  - e. Lista Circular
  - f. Lista Duplamente Encadeada (Dupla Circular)
- 2) Como poderemos achar uma celebridade em uma multidão, onde todos conhecem a celebridade (que não conhece ninguém da multidão)?
- 3) Como podemos saber qual o candidato vence por maioria absoluta (50% + 1 voto) uma eleição?
- 4) Como nosso celular encontra o nome de um contato armazenado na agenda? (dica: a lista de nomes está ordenada)
- 5) Implementar os algoritmos de ordenação:
  - a. Insertion Sort
  - b. Selection Sort
  - c. Bubble Sort
  - d. Quick Sort
  - e. Merge Sort
- 6) Implementar o algoritmo da Torre de Hanoi.
- 7) Implementar um algoritmo que, dado duas listas com os coeficientes de dois polinômios, retorne a soma do polinômio com seus coeficientes em uma lista.

- 8) Desenvolva um algoritmo que determine se um número  $N$  informado é primo e informe os dois primos antecessores e os dois primos sucessores de  $N$ .
- 9) Duas empresas abriram falência, onde cada empresa possui o mesmo grupo de credores que estão ávidos para ressarcirem seus prejuízos. A empresa P1 foi avaliada em R\$  $X,00$ . A empresa P2 foi avaliada em R\$  $Y,00$ . Note que tanto  $X$  e  $Y$  são valores inteiros e positivos, diferentes de zero. O juiz responsável pela dissolução das empresas determina que o maior número possível de credores ( $Z$ ) deverá receber o mesmo valor fixo, de cada empresa, onde  $Z$  é um valor inteiro e positivo, diferente de zero, se constituindo no maior divisor comum entre os valores de avaliação associados a cada empresa. Ou seja, cada credor receberá  $X/Z + Y/Z$ . Descreva um algoritmo que especifique o número máximo de credores ( $Z$ ) que serão beneficiados com a decisão do judiciário.
- 10) Uma das formas de encontrar o número de Fibonacci é através de um laço de repetição ou através de chamadas recursivas. O segundo caso é ilustrado a seguir, apresentando a árvore de derivação ao calcularmos o valor  $\text{fib}(4)$ , ou seja, o sexto valor dessa sequência:

$\text{fib}(0) = 0$ ,  $\text{fib}(1) = 1$ ,  $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$



Implemente:

- A versão não-recursiva para encontrar o número da  $n$  da sequência de Fibonacci.
  - A versão recursiva que retorne a **quantidade de chamadas recursivas** feitas (no exemplo acima, temos 8 chamadas).
- 11) Implementar um algoritmo para identificar a quantidade de palíndromos presentes em um texto. Os palíndromos podem ser palavras, frases ou o próprio texto. Utilize para separar as frases a pontuação de vírgula, ponto ou ponto final, pontos de exclamação e pontos de interrogação.

- 12)** Desenvolva um algoritmo que faça a conversão da notação infixa para a notação polonesa (prefixa).
- 13)** Desenvolva um algoritmo que faça a conversão da notação infixa para a notação polonesa reversa (pósfixa).
- 14)** Desenvolva um algoritmo que faça a conversão da notação polonesa (prefixa) para a notação infixa.
- 15)** Desenvolva um algoritmo que faça a conversão da notação polonesa reversa (pósfixa) para a notação infixa.