UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá
Prof. Arthur Araruna
QXD0115- Estrutura de Dados Avançada

LE1
2025.1

Nome: _____ Matrícula: _____

Observação
Para os exercícios a seguir, quando alguma questão solicitar um algoritmo que recebe ou retorna uma árvore, considere que essa árvore será sempre representada por seu nó raiz.

1 Exercícios de Fixação

1. Escreva um algoritmo que retorna o tamanho de uma árvore a partir do seu nó raiz.
2. Dada uma árvore binária T e um valor x, retornar uma lista contendo todos os ancestrais de x em T ordenados por nível (do mais alto para o mais baixo na árvore).
3. Dadas duas árvores binárias T_1 e T_2, escreva um algoritmo que determina se essas árvores são iguais. Ou seja, se possuem o mesmo desenho e se os nós correspondentes possuem os mesmos valores.
4. Escreva os percursos em Pré-Ordem, Em-Ordem e Pós-Ordem da árvore binária na Figura 1.

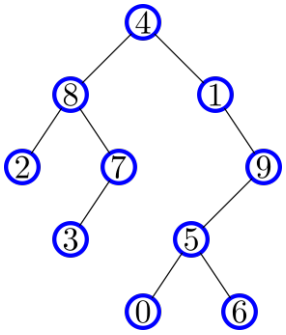


Figura 1:

5. Se implementarmos uma árvore binária em uma linguagem que não gerencia automaticamente a memória solicitada e em algum momento nos depararmos com a necessidade de nos desfazer de uma árvore, como devemos proceder? Explique em forma de um algoritmo, supondo que o algoritmo que efetivamente retorna um espaço de memória ao sistema se chame DevolveNó(n).
6. **POSCOMP 2015. (adaptada)**
Considere T uma árvore binária cheia, em que n, n_e, n_i e h representam o número de nós, o número de nós externos (sub-árvores vazias), o número de nós internos (demais nós) e a altura de T, respectivamente. Portanto, a essa árvore T aplica-se a seguinte propriedade:
   - ◯ $n_i = n_e + 1$
   - ◯ $h - 1 \leq n_e \leq 2^h$
   - ◯ $h + 1 \leq n_i \leq 2^h$
   - ◯ $\lg(n+1) \leq h \leq n - 1$
   - ◯ $2h + 1 \leq n \leq 2^{h+1} - 1$
7. Considere os algoritmos de Sucessor e Predecessor de cada um dos percursos estudados e a árvore binária representada na Figura 2 na próxima página. Para cada um dos item a seguir, liste a sequência de nós pelos quais o algoritmo correspondente passa a partir do nó de valor informado até o nó desejado.

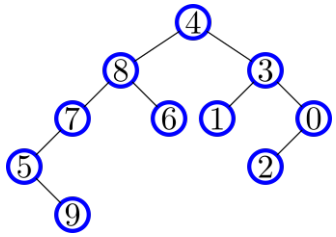


Figura 2:

(a) Sucessor-Pré(9)
(b) Sucessor-Pré(2)
(c) Sucessor-Pós(8)
(d) Antecessor-Em(7)
(e) Antecessor-Em(1)
(f) Antecessor-Pós(3)

8. Para cada item a seguir, exiba uma *árvore binária* (não necessariamente de busca) que gere as mesmas sequências de nós percorridos que os percursos informados.

   OBS: Uma árvore para cada item. Ambos os percursos válidos para essa árvore.

   (a) **Pré-Ordem:** $\langle 10,7,8,4,1\rangle$ **Pós-Ordem:** $\langle 8,4,7,1,10\rangle$
   (b) **Pré-Ordem:** $\langle 3,4,5,6\rangle$ **Pós-Ordem:** $\langle 6,5,4,3\rangle$
   (c) **Pré-Ordem:** $\langle 8,4,10,12,15\rangle$ **Pós-Ordem:** $\langle 10,4,15,12,8\rangle$
   (d) **Pré-Ordem:** $\langle 13,12,10,9,8,11,7,6\rangle$ **Pós-Ordem:** $\langle 10,8,9,12,6,7,11,13\rangle$
   (e) **Pós-Ordem:** $\langle 4,1,8,7,5\rangle$ **Em-Ordem:** $\langle 4,8,1,5,7\rangle$
   (f) **Pós-Ordem:** $\langle 1,6,5,3,2\rangle$ **Em-Ordem:** $\langle 1,2,5,6,3\rangle$
   (g) **Pós-Ordem:** $\langle 1,5,4,2,3\rangle$ **Em-Ordem:** $\langle 1,2,5,4,3\rangle$
   (h) **Pós-Ordem:** $\langle 2,1,3,6,7,5,4\rangle$ **Em-Ordem:** $\langle 2,4,1,3,5,6,7\rangle$

2 Exercícios de Aplicação

9. Dada uma árvore binária em que cada nó possui uma referência para o seu pai ainda não preenchida, escreva um algoritmo para realizar corretamente seu preenchimento.

10. Escreva um algoritmo que, dada uma árvore binária T, retorna um clone de T. Ou seja, uma árvore T' com a mesma estrutura e os mesmos valores nos nós correspondentes, porém sem aproveitar os nós existentes em T.

    OBS: A versão recursiva deste algoritmo é mais simples.

11. Observe o Algoritmo 1. Suponha que r é um nó de uma árvore binária.

Algoritmo 1:

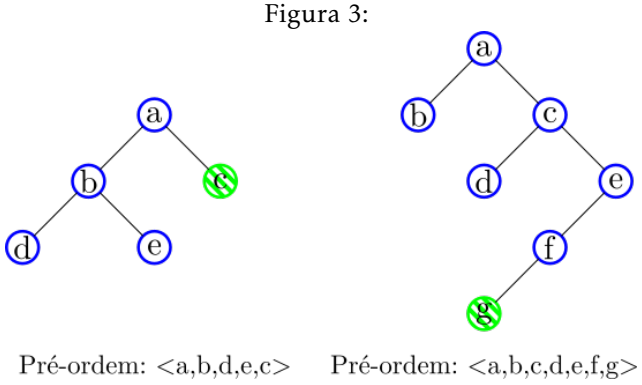
```
E ← ∅
Inserir(E, r)
enquanto E ≠ ∅ faça
    n ← Remover(E)
    Imprime(n.chave)
    se n.esq ≠ λ então
        Inserir(E, n.esq)
    se n.dir ≠ λ então
        Inserir(E, n.dir)
```

(a) Considere que E seja uma *fila*. A ordem em que os nós são impressos pelo algoritmo guarda alguma propriedade interessante? Qual?

(b) Considere que E seja uma *pilha*. A ordem em que os nós são impressos pelo algoritmo guarda alguma propriedade interessante? Qual?

12. Chamamos de *máximo em pré-ordem* de uma árvore binária T o último nó de T que é listado por um percurso em pré-ordem. Observe as árvores na Figura 3, onde são exibidos seus percursos em pré-ordem, com destaque para o nó máximo.

Figura 3:



Escreva um algoritmo que, dado T, retorna o seu máximo em pré-ordem sem necessitar percorrer todos os nós da árvores para isso.

OBS: Em sala, chamamos este algoritmo de ÚltimoPré.

13. Imagine que em cada nó de uma árvore binária T acrescentamos três campos, `pre`, `pos` e `em`, de forma que o valor de cada campo representa a posição que o nó em questão ocupa em um percurso em pré-ordem, pós-ordem e em-ordem, respectivamente. Assim, se o valor de `n.pré` for 1, isso significa que `n` é o primeiro nó de T ao percorrê-la em pré-ordem. Mostre como usar essas novas informações para responder cada uma das situações a seguir em tempo constante:
    (a) Dado um nó n, determinar o número de nós na sub-árvore enraizada em n.
    (b) Dado um nó n, determinar o nível de n.
    (c) Dados dois nós n e m, determinar se n é ancestral de m.

14. Se incluíssemos também as sub-árvores vazias (λ) na saída do percurso em pré-ordem de uma árvore, seríamos capazes de determinar unicamente a árvore original? Mostre como seria o desenho da árvore cujo percurso é $\langle 15, 0, 10, \lambda, \lambda, 20, \lambda, \lambda, 25, \lambda, 5, \lambda, \lambda \rangle$ e indique onde há ambiguidade, caso houver.

OBS: Uma folha é um nó cujas duas sub-árvores são vazias.

3 Desafios

15. Dados dois vetores `pre` e `em`, representando respectivamente um percurso em pré-ordem e um em-ordem de uma mesma árvore binária T, escreva um algoritmo para recriar T e retornar sua raiz.
    (a) Escreva uma versão recursiva.
    (b) Escreva uma versão iterativa.

16. Escreva um algoritmo que, dados uma árvore binária T (qualquer, não necessariamente de busca) e dois nós n_1 e n_2, ambos em T, retorne o nó ancestral comum a n_1 e n_2 que esteja no nível mais baixo. Veja um exemplo na Figura 4 na página seguinte, onde o nó desejado encontra-se destacado.

17. Dada uma árvore binária T, cujos nós armazenam números inteiros, e um valor k, escreva um algoritmo que descobre se em T existe alguma sequência de nós que some exatamente k. Essa sequência pode iniciar em qualquer nó, mas não deve saltar nenhum nó ao descer na árvore.

18. Dada uma árvore binária T, escreva um algoritmo para determinar se em T existem sub-árvores duplicadas. Veja um exemplo na Figura 5 na próxima página.

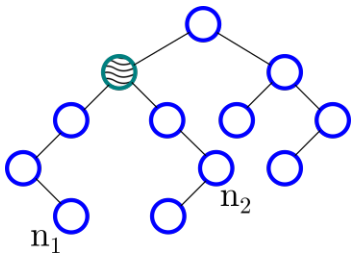


Figura 4:

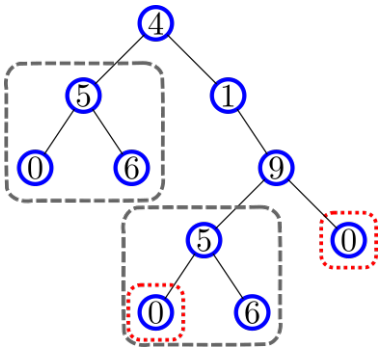


Figura 5:

19. Dada uma árvore binária T, escreva um algoritmo que detecta se T é Binária de Busca. Que informação seria útil informar ao algoritmo (ou incluir nos retornos) de forma a evitar descer na árvore mais de uma vez?

20. Dada uma ABB cheia T e um valor $x \in T$, imaginando que os nós foram numerados da esquerda para a direita, um nível por vez de cima para baixo, determinar em $O(h)$ qual foi o número associado ao nó que contém x.