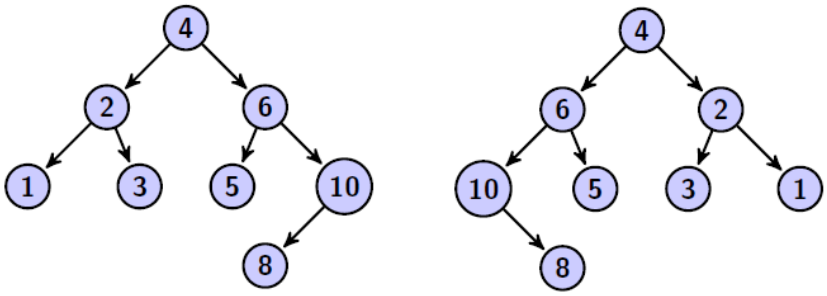
ESTRUTURA DE DADOS

Exercícios: Árvores

1. Utilizando os conceitos de grafos, defina uma árvore.
2. Escreva uma função que conta o número de nós de uma árvore binária.
3. Escreva uma função que conta o número de nós não-folha de uma árvore binária.
4. Escreva uma função que conta o número de folhas de uma árvore binária.
5. Escreva uma função que calcula a altura de uma árvore binária.
6. Escreva uma função para buscar um número impar em uma árvore binária NÃO ordenada.
7. Escreva uma função para achar o MAIOR número em uma árvore binária NÃO ordenada.
8. Escreva uma função que exclui todos os nós de uma árvore binária NÃO ordenada com ID par.
9. Escreva uma função que exclui todos os nós de uma árvore binária de busca com ID par.
10. Escreva uma função que retorna **verdadeiro** se uma árvore é binária de busca e **falso** caso contrário
11. Escreva uma função que encontra o valor máximo em uma árvore de busca binária.
12. Escreva uma função que obtém o espelho de uma árvore, ou seja, troca a subárvore direita pela subárvore esquerda de todos os nós da árvore



13. Duas ABBs são SIMILARES se possuem a mesma distribuição de nós (independente dos valores nos mesmos). Em uma definição mais formal, duas ABBs são SIMILARES se são ambas vazia, ou se suas subárvores esquerdas são similares, e suas subárvores direitas também são similares. Implemente a função que verifica se duas árvores são similares.
14. Duas ABBs são IGUAIS se são ambas vazias ou então se armazenam valores iguais em suas raizes, suas subárvores esquerdas são iguais, e suas subárvores direitas são iguais. Implemente a função que verifica se duas árvores são simlares.

15. Uma ABB é estritamente binária se todos os nós da árvore tem 2 filhos. Implemente uma função que verifica se uma ABB é estritamente binária.

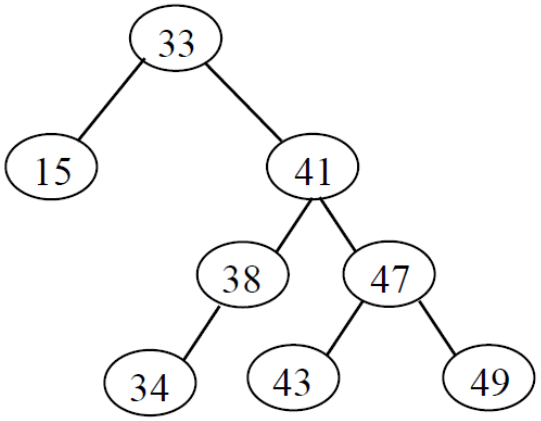
16. Implemente uma função para testar se uma árvore binária é uma ABB.

17. Pense na implementação não recursiva dos algoritmos de inserção, remoção e busca em uma ABB.

18. Dada uma ABB inicialmente vazia, insira (E DESENHE) os seguintes elementos (nessa ordem): M, F, S, D, J, P, U, A, E, H, Q, T, W, K.

19. Dada uma ABB inicialmente vazia, insira (E DESENHE) os seguintes elementos (nessa ordem): A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

20. Descreva a ordem de visita para um percurso em pré-ordem, em-ordem e pós-ordem na árvore abaixo



21. Qual a diferença de uma ABB para uma AVL?

22. O que difere na implementação da função de busca de uma ABB e de uma AVL?

23. Obtenha a equação que relaciona a altura de uma árvore binária completa com o seu número de vértices.

24. Escreva funções não recursivas para realizar os 3 tipos de percurso na árvore binária (dica: use uma pilha):

    (a) pré-ordem
    (b) em-ordem
    (c) pós-ordem

25. Escreva uma função não-recursiva que retorne o menor valor de uma árvore binária (não ordenada)

26. Escreva uma função não-recursiva que retorne o menor valor de uma árvore binária de busca.

27. Escreva uma função não-recursiva que retorne o k-ésimo menor valor de uma árvore binária de busca.

28. Escreva uma função que retorne o elemento menor ou igual (floor) que um de referência **x** em uma ABB.

29. Escreva uma função não-recursiva que verifique a existência de um valor X na árvore binária.

30. Escreva uma função não-recursiva que verifique a existência de um valor negativo na árvore binária.

31. Escreva uma função não-recursiva que verifique se uma árvore binária é também de busca.

32. Mostre passo a passo a árvore binária resultante das seguintes operações:

    (a) Inserção de 7, 8, 3, 4, 2, 1, 6, 5

    (b) Mostre o percurso em pré-ordem, em-ordem e pós-ordem

    (c) Remoção de 7 e 6

33. Escreva e implemente um algoritmo não recursivo para obter a altura de uma ABB.

34. Escreva e implemente um algoritmo que dada uma ABB, construa uma outra árvore ABB aproximadamente completa. Para isso, obtenha todas as chaves e valores e insira na nova árvore sempre o elemento mediano das chaves ainda não inseridas.

35. Faça uma função que retorne a quantidade de folhas de uma árvore binária de busca.

36. Faça uma função que retorne a quantidade de nós de uma árvore binária de busca que possuem apenas um filho.

37. Faça uma função que, dada uma árvore binária de busca, retorne a quantidade de nós que guardam números primos.

38. Faça uma função que compare se duas árvores binárias de busca são iguais.

39. Faça uma função que retorna a lista de caminhos da raiz até cada folha.

40. Escreva uma função que faça o percurso em nível em uma árvore binária. O percurso em nível (ou em largura) em uma árvore é um percurso que visita, em ordem crescente, todos os nós de um nível antes de continuar a visita no nível seguinte. Uma das formas de implementar é utilizar uma fila (FIFO) para guardar quais serão os próximos nodes a serem visitados.

41. Considere a árvore rubro-negra caida para a esquerda cujo percurso em nível (em largura) é: 67 51 87 23 53 82 90 17 31 52 60 16 21. Liste as chaves em nós rubros em ordem crescente.

42. Defina com suas palavras o que é uma árvore AVL e como ela funciona.

43. Explique as vantagens e desvantagens de usar árvores binárias balanceadas?

44. Desenhe a árvore AVL resultante da inserção dos seguintes nós: 35, 39, 51, 20, 13, 28, 22, 32, 25, 33 (nesta ordem).

45. Qual a diferença de uma árvore binária de busca para uma AVL?

46. Uma árvore binária de busca é estritamente binária se todos os nós da árvore tem 2 filhos. Implemente uma função que verifica se uma árvore binária de busca é estritamente binária.

47. Escreva um programa para copiar uma árvore binária.

48. Faça um esquema gráfico que represente a estrutura final de uma árvore AVL gerada pela inserção dos números 13, 7, 31, 43, 8, 17, 5, 3, 1, 23, 16 e 36 nesta ordem

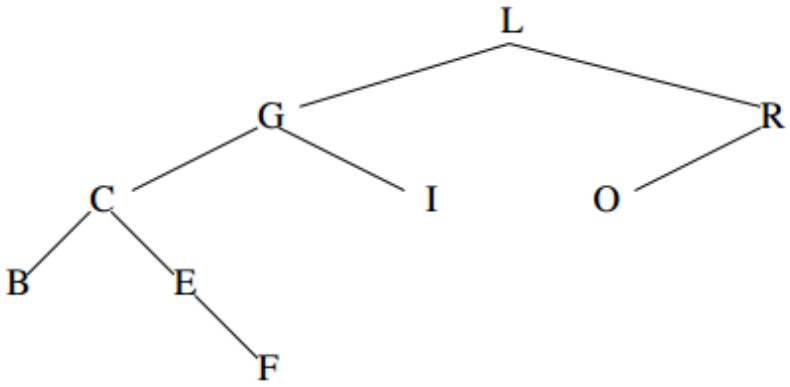
49. Modifique a implementação da árvore de busca binária para que ela lide corretamente com chaves duplicadas. Isto é, se uma chave já está na árvore, a nova dever´substituir a antiga.

50. Exiba as árvores binárias com alturas de 2, 3, 4, 5 e 6, que contenham as sete chaves: 1, 4, 5, 10, 16, 17 e 21. Eemplo: com altura de 2:

```
  10
 /  \
  4   17
 / \  / \
1  5 16 21
```

51. Considere a árvore mostrada na figura abaixo e responda:

    (a) Quais são os nós folhas?
    (b) Quais nós são ancestrais de C?
    (c) Quais são os descendentes de C?
    (d) Qual é a altura da árvore?
    (e) Quais são os nós com grau 1 e 2?
    (f) Quantos caminhos de comprimento três existem?



52. Crie uma árvore binária de busca, adicionando um a um, e nesta ordem, os seguintes números: 5, 6, 3, 8, 7, 4, 1 e 2.

53. Quantas árvores de busca binária com nós 1, 2, 3 e 4 existem?

54. Considere uma árvore de busca binária tendo como nós os números de 1 até 1000. Nós procuramos um nó marcado com 363. Qual das seguintes sequências não podem ser os valores dos nodos visitados na busca por 363?

    (a) 2; 252; 401; 398; 330; 344; 397; 363.
    (b) 924; 220; 911; 244; 898; 258; 362; 363.
    (c) 925; 202; 911; 240; 912; 245; 363.
    (d) 2; 399; 387; 219; 266; 382; 381; 278; 363.
    (e) 935; 278; 347; 621; 299; 392; 358; 363.

55. Dê um pequeno exemplo de como adicionar um número a uma árvore AVL causando uma árvore desbalanceada do tipo direita direita. Rebalancear a árvore.

56. Dê um pequeno exemplo de como adicionar um número a uma árvore AVL causando uma árvore desbalanceada do tipo direita-esquerda. Rebalancear a árvore.

57. Dê um pequeno exemplo de remoção de um número de um árvore AVL causando uma árvore desbalanceada do tipo direita-direita. Rebalancear a árvore.

58. Projete um algoritmo para determinar o menor número em uma árvore AVL.

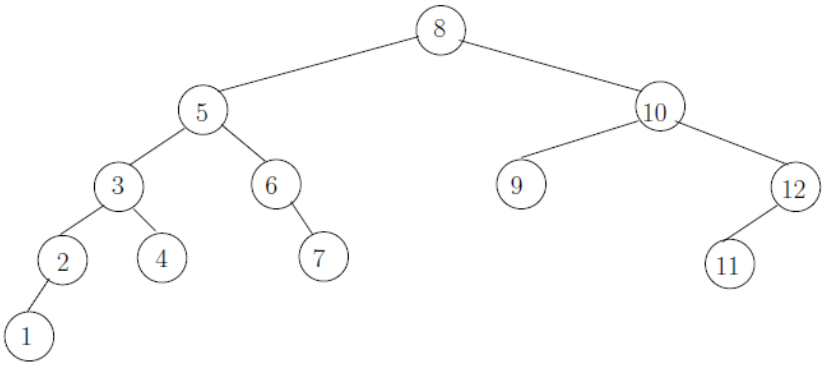
59. Escreva uma função que receba um nível da árvore e imprima todos os nós nesse nível.

60. Escreva a função que retorna o número de nós que são divisores da soma de seus filhos. Leve em conta apenas os nós com dois filhos.

61. Dada uma árvore binária e dois nós dela, desenvolva um algoritmo para achar o ancestral comum mais baixo dos dois nós dados. O ancestral comum é um nó que possua os outros dois nós em suas subárvores. O ancestral comum mais baixo é o ancestral comum que tem a maior profundidade.

62. Em uma árvore binária, faça um algoritmo para alterar o valor de cada nó (exceto nó folha) para a soma dos valores dos nós esquerda e direita.

63. Remova da seguinte árvore AVL o nó 8 e, em seguida, adicione um nó com valor 9.



64. Faça um algoritmo que crie uma lista ligada com os nós de uma árvore binária em um percurso em-ordem.

65. Faça um algoritmo para somar os nós presentes nos níveis impares de uma árvore binária

66. Implementar uma função não-recursiva para calcular o tamanho de uma árvore binária.

67. Implementar uma função não-recursiva para cada um dos percursos abaixo

    (a) em-ordem

    (b) pós-ordem

    (c) pré-ordem

68. Monte uma árvore capaz de armazenar uma struct contendo as seguintes informações: nome do município, área total do município (em km2) e população. A árvore deverá ter funções para inserir (pelo nome), percorrer e listar. Implemente também funções para:

(a) Contar o número de municípios, percorrendo os nós cadastrados na árvore

(b) Mostrar apenas os nomes dos municípios com mais de X habitantes. Por exemplo, X pode ser 100.000 pessoas.

(c) Mostrar a densidade demográfica de cada cidade. A densidade demográfica é a relação entre a população e a área.

(d) Mostrar o somatório de área em km2 de todas as cidades juntas em relação ao território nacional (em porcentagem).

(e) Mostrar as cidades em ordem alfabética, com todos os dados.

(f) Mostrar o nome da cidade com a maior população.