



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga  
Ciência da Computação – Estrutura de Dados e Algoritmos  
Prova III – 2º/2018 – Árvores  
Prof. Daniel Saad Nogueira Nunes

Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_

Data: 6 de dezembro de 2018

Duração da prova: 100 minutos
-------------------------------

Tabela de notas (uso exclusivo do professor)

Questão	Pontos	Nota
1	3	
2	3	
3	3	
4	3	
Total	12	

## Observações

- Esta prova tem o total de 2 páginas (incluindo a capa) e 4 questões.
- O número total de pontos é 12.
- Certifique-se de assinar todas as folhas de resposta bem como a capa da prova.
- Leia atentamente todas as questões da prova. A interpretação do problema é crucial para o desenvolvimento correto da resposta.
- Resoluções sem justificativa não serão consideradas.
- É vedado o uso de equipamentos eletrônicos, como celulares, notebooks entre outros.
- A prova será **anulada** e medidas disciplinares serão tomadas para os alunos que “colarem” durante a avaliação.

★ Certifique-se de assinar todas as folhas de resposta.

---

### Questão 1 (3 pontos)

Crie uma função que recebe duas árvores binárias do mesmo tipo e retorne verdadeiro se elas são iguais e falso caso contrário. Sua função deverá possuir a seguinte assinatura.

```
int compare_tree(tree_t* t1, tree_t* t2);
```

### Questão 2 (3 pontos)

De acordo com percurso em árvores binárias:

- (a) (1½ pontos) Implemente a busca em profundidade em pré-ordem. Sua função deverá possuir a seguinte assinatura:

```
void busca_profundidade(tree_t* t);
```

- (b) (1½ pontos) Implemente a busca em largura. Sua função deverá possuir a seguinte assinatura:

```
void busca_largura(tree_t* t);
```

### Questão 3 (3 pontos)

Considerando árvores binárias de pesquisa:

- (a) (1½ pontos) Escreva uma função, com assinatura dada abaixo, que realize a inserção de um dado na BST.

```
void bst_insert(bst_t* t, void* data);
```

- (b) (1½ pontos) Escreva uma função, com assinatura dada abaixo, que realize a remoção de um dado na BST, caso ele esteja na árvore.

```
void bst_remove(bst_t* t, void* data);
```

### Questão 4 (3 pontos)

No âmbito de árvores AVL:

- (a) (1½ pontos) A partir de uma árvore AVL vazia, informe e ilustre a sequência de operações realizadas para deixá-la balanceada mediante a inserção dos elementos (10, 77, 75, 7, 5, 3, 4), nesta ordem. Para cada nó, você deverá indicar qual é o fator de equilíbrio a cada operação que é realizada.

- (b) (1½ pontos) Implemente as rotações para a esquerda e para direita em árvores AVL. Essas rotações agem sobre a raiz da subárvore considerada e devem retornar a nova raiz. Suas funções devem possuir as seguintes assinaturas:

```
avl_node_t* avl_left_rotate(avl_node_t* x);  
avl_node_t* avl_right_rotate(avl_node_t* x);
```

★ Certifique-se de assinar todas as folhas de resposta.