

Disciplina: Algoritmos e Estruturas de Dados I (EDI)

Professor: Eduardo de Lucena Falcão

Exercício sobre Árvores

1) Responda às seguintes perguntas:

- Qual a diferença entre uma árvore binária e uma árvore de busca binária (BST)?
- O que significam os níveis de uma árvore?
- O que é uma BST completa?
- Como calculamos o fator de balanceamento de uma árvore?
- Qual a altura de uma árvore balanceada contendo n nós? Mostre matematicamente.
- Qual a complexidade de tempo das seguintes operações em uma árvore completamente desbalanceada? Add(value int), Search(value int) bool, Min() int, Max() int, PrintPre(), PrintIn(), PrintPos(), Height() int, Remove(value int) *BstNode
- Qual a complexidade de tempo das seguintes operações em uma árvore completa? Add(value int), Search(value int) bool, Min() int, Max() int, PrintPre(), PrintIn(), PrintPos(), Height() int, Remove(value int) *BstNode
- Explique os possíveis casos de remoção em uma BST. Como deve-se proceder em cada caso?

2) Programe uma Binary Search Tree na linguagem Go.

```
type ITree interface {
    Add(value int)
    Search(value int) bool
    Min() int
    Max() int
    PrintPre()
    PrintIn()
    PrintPos()
    PrintLevels()
    Height() int
    Remove(value int) *BstNode
}

type BstNode struct {
    left  *BstNode
    value int
    right *BstNode
}
```

3) Considere uma Árvore de Busca Binária que foi criada através da inserção dos 10 primeiros caracteres do seu nome (sem repetição). (2.0)

—, —, —, —, —, —, —, —, —, —.

- Qual seria a ordem de visitação/processamento ao aplicar uma navegação Pré-Ordem?

- b) Qual seria a ordem de visitação/processamento ao aplicar uma navegação Em-Ordem?
- c) Qual seria a ordem de visitação/processamento ao aplicar uma navegação Pós-Ordem?
- d) Qual seria a ordem de visitação/processamento ao aplicar uma navegação por níveis?

- 4) Escreva uma função que identifica se uma árvore é uma BST.

```
type BstNode struct {  
    left  *BstNode  
    value int  
    right *BstNode  
}  
  
func (bstNode *BstNode) isBst() bool
```

- 5) Escreva uma função que retorna a quantidade de elementos de uma BST.

```
type BstNode struct {  
    left  *BstNode  
    value int  
    right *BstNode  
}  
  
func (bstNode *BstNode) Size() int
```

- 6) Escreva uma função que recebe um vetor contendo elementos desordenados e retorna uma BST balanceada.

```
type BstNode struct {  
    left  *BstNode  
    value int  
    right *BstNode  
}  
  
func createBst(v []int) *BstNode
```