Teórico 8

Gustavo Lopes Rodrigues

6 de Maio de 2020

1 Questão 1

```
1 public int soma(){
2   return soma(this.raiz);
3 }
4
5 public int soma(No i){
6   int resp = 0;
7
8     if(i != null){
9
10          resp = i.elemento + soma(i.esq) + soma(i.dir);
11
12    }
13
14  return resp;
15 }
```

2 Questão 2

```
1 public int contPares(){
2       return contPares(this.raiz);
3 }
4
5 public int contPares(No i){
6   int resp = 0;
```

```
7
8       if(i != null){
9
10          resp = ((i.elemento % 2 == 0) ? 1 : 0) + contPares(i.esq) +
        contPares(i.dir);
11
12      }
13
14      return resp;
15  }
```

3 Questão 3

```
1  public static boolean igual (ArvoreBinaria a1, ArvoreBinaria a2){
2      return igual(a1.getRaiz(), a2.getRaiz());
3  }
4
5  private static boolean igual (No i1, No i2){
6      boolean resp;
7
8      if(i1 != null && i2 != null){
9
10         resp = (i1.elemento == i2.elemento) && igual(i1.esq, i2.esq)
        && igual(i1.dir, i2.dir);
11
12     }
13   else if(i1 == null && i2 == null){
14
15         resp = true;
16
17     }
18     else {
19
20         resp = false;
21
22     }
23
24     return resp;
```

```
}
```

4 Questão 4

```
public boolean temDivisor11(){
    return temDivisor11(this.raiz);
}

public boolean temDivisor11(No i){
    boolean resp = false;

    if(i != null){

       resp = (i.elemento % 11 == 0) || temDivisor11(i.esq) ||
    temDivisor11(i.dir);

    }

    return resp;
}
```

5 Questão 5

```
public No toArvoreBinaria(Celula i,CelulaDupla j) throws Exception{
  No resp = null;

    if ( i.prox != null && j.prox != null ) {

      resp = toArvoreBinaria(resp,i.prox,j.prox);

    }

  return resp;
}

public No toArvoreBinaria(No resp,Celula i,CelulaDupla j) throws
    Exception {

```

```
15   if(i != null) {
16
17     resp = inserir(i.elemento,resp);
18     i = i.prox;
19
20   }
21   if(j != null) {
22
23     resp = inserir(j.elemento,resp);
24     j = j.prox;
25
26   }
27   if ( i != null || j != null ) {
28
29     resp = toArvoreBinaria(resp,i,j);
30
31   }
32
33   return resp;
34
35 }
```

6 Questão 6

```
1   public void inserir(int x) throws Exception {
2       if(raiz == null){
3          raiz = new No(x);
4       } else if(x < raiz.elemento){
5        inserir(x, raiz.esq, raiz);
6       } else if(x > raiz.elemento){
7        inserir(x, raiz.dir, raiz);
8       } else {
9          throw new Exception("Erro ao inserir2!");
10      }
11  }
12
13  private void inserir(int x, No i, No pai) throws Exception {
14    if (i == null) {
```

```
15          if(x < pai.elemento){
16             pai.esq = new No(x);
17          } else {
18             pai.dir = new No(x);
19          }
20       } else if (x < i.elemento) {
21          inserir(x, i.esq, i);
22       } else if (x > i.elemento) {
23          inserir(x, i.dir, i);
24       } else {
25          throw new Exception("Erro ao inserir2!");
26       }
27    }
```

7 Questão 7

```
1   public void remover(int x) throws Exception {
2       if (raiz == null) {
3          throw new Exception("Erro ao remover2!");
4       } else if(x < raiz.elemento){
5          remover(x, raiz.esq, raiz);
6       } else if (x > raiz.elemento){
7          remover(x, raiz.dir, raiz);
8       } else if (raiz.dir == null) {
9          raiz = raiz.esq;
10      } else if (raiz.esq == null) {
11         raiz = raiz.dir;
12      } else {
13         raiz.esq = antecessor(raiz, raiz.esq);
14      }
15   }
16
17  private void remover(int x, No i, No pai) throws Exception {
18    if (i == null) {
19         throw new Exception("Erro ao remover2!");
20      } else if (x < i.elemento) {
21         remover(x, i.esq, i);
22      } else if (x > i.elemento) {
```

```
23          remover(x, i.dir, i);
24       } else if (i.dir == null) {
25      if ( pai.esq == i ) {
26       pai.esq = i.esq;
27      }
28      else {
29       pai.dir = i.esq;
30      }
31       } else if (i.esq == null) {
32      if ( pai.esq == i ) {
33       pai.esq = i.dir;
34      }
35      else {
36       pai.dir = i.dir;
37      }
38       }  else {
39          i.esq = antecessor(i, i.esq);
40     }
41  }
```