

Universidade Estadual de Maringá - Ciência Da Computação (DIN)  
Aluno: Otávio Hideki Gonçalves Kochi RA:107635  
Aluno: Thiago Issao Yasunaka RA:103069

## **Trabalho1 – Algoritmo em Grafos 6898**

### **Árvores Aleatórias**

Parte 1 – Função diametro(T)

IMPLEMENTAÇÃO UTILIZANDO A LINGUAGEM JAVASCRIPT

Repositório no GitHub:

[https://github.com/thiagoissao/arvore\\_aleatorias\\_grafos/blob/master/diametro.js](https://github.com/thiagoissao/arvore_aleatorias_grafos/blob/master/diametro.js)

### **Código:**

```
const assert    = require('assert')
const BRANCO    = 'branco'
const CINZA     = 'cinza'
const PRETO     = 'preto'

const diametro = (G) => {
  const s = Math.round(Math.random() * (G.length - 1))
  const a = bfs(G, s).reduce((number1, number2) => {
    return Math.max(number1, number2)
  })
  const b = bfs(G,a).reduce((number1, number2) => {
    return Math.max(number1, number2)
  })
  return a > b ? a:b
}

const teste_diametro = () => {
  let a = 0
  let b = 1
  let c = 2
  let d = 3
  let e = 4
  let f = 5
  let g = 6
  let h = 7
  const G = [
```

```

    [b,c],          //      a
    [a],            //      /  \
    [a]             //     b    c
  ]
  const G2 = [
    [a]             //      a
  ]
  const G3 = [
    [b,c],          //      a
    [a,e,d],        //      /  \
    [a,f],           //      b    c
    [b],             //      /\    \
    [b],             //      e    d    f
    [c,g],           //                \
    [f],             //                g
  ]
  let dis = diametro(G)
  let dis2 = diametro(G2)
  let dis3 = diametro(G3)
  assert (dis == 2)
  assert (dis2 == 0)
  assert (dis3 == 5)
}

```

```

const bfs = (G, s) => {
  let GAux = Object.assign({}, G)
  Object.keys(GAux).forEach( index => {
    let adj = G[index]
    GAux[index] = {
      'd': Number.POSITIVE_INFINITY,
      'pi': null,
      'cor': BRANCO,
      'adj': adj
    }
  })
  GAux[s].d = 0
  GAux[s].pi = null
  GAux[s].cor = CINZA
  let pilha = [] //Inicialização da pilha
  pilha.push(GAux[s])
  while (pilha.length != 0) {

```

```

    let u = pilha.pop()
    u.adj.forEach( v => {
        if(GAux[v].cor === BRANCO) {
            GAux[v].d = u.d + 1
            GAux[v].pi = G.indexOf(u.adj)
            GAux[v].cor = CINZA
            pilha.push(GAux[v])
        }
    })
    u.cor = PRETO
}
return Object.keys(GAux).map(index => GAux[index].d)
}

```

```

const teste_bfs = () => {
    let r = 0
    let s = 1
    let t = 2
    let u = 3
    let v = 4
    let w = 5
    let x = 6
    let y = 7
    const G = [
        [s,v],
        [r,w],
        [u,w,x,y],
        [t,x,y],
        [r],
        [s,t,x],
        [w,t,u,y],
        [u,x]
    ]
    const G2 = [
        [s,v],           //    r
        [r,t,u],         //    /\
        [w,s],           //    v  s
        [s],             //    /\
        [r],             //    t  u
        [t]              //    /
    ]                  //    w
    let d = bfs(G, 1)
}

```

```
let d1 = bfs(G2, 0)
```

```
assert (d[r] == 1)
```

```
assert (d[s] == 0)
```

```
assert (d[t] == 2)
```

```
assert (d[u] == 3)
```

```
assert (d[v] == 2)
```

```
assert (d[w] == 1)
```

```
assert (d[x] == 2)
```

```
assert (d[y] == 3)
```

```
assert(d1[r] == 0)
```

```
assert(d1[s] == 1)
```

```
assert(d1[t] == 2)
```

```
assert(d1[u] == 2)
```

```
assert(d1[v] == 1)
```

```
assert(d1[w] == 3)
```

```
}
```

```
teste_bfs()
```

```
teste_diametro()
```