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)

IMPLEMENTACÃO UTILIZANDO A LINGUAGEM JAVASCRIPT

Repositório no GitHub:
https://github.com/thiagoissao/arvore_aleatorias_grafos/blob/master/diametro.js

Código:

```
const numero_arestas = (G) => {
    let contador = 0;
    if(G.length > 1)
        for(let i = 0; i < G.length; i ++){
            for(let j = 0; j < G[i].length; j ++){
                if(G[i][j] <= i)
                    contador++;
            }
        }
    return contador;
}

const teste_aresta = () => {
    const G4 = [
        [b,c],                  //       a
        [a],                    //     /   \
        [a]                     //   b      c
    ]
    const G = [
        [a]
    ]
    const G2 = [
        [s,v],                  //      r
        [r,t,u],                //     / \
        [w,s],                  //    v   s
        [s],                    //        / \
        [r],                    //       t    u
        [t]                     //     /
    ]                           //   w
    const G3 = [
```

```
        [b,c],                  //          a
        [a,e,d],                //         /  \
        [a,f],                  //        b    c
        [b],                    //       / \     \
        [b],                    //      e   d    f
        [c,g],                  //                \
        [f],                    //                 g
    ]
    const are1 = numero_arestas(G)
    const are2 = numero_arestas(G2)
    const are3 = numero_arestas(G3)
    const are4 = numero_arestas(G4)
    assert(are1 == 0)
    assert(are2 == 5)
    assert(are3 == 6)
    assert(are4 == 2)
}

const random_tree_random_walk = n => {
    let GAux = Object.assign({}, Array.apply(null, Array(n)))
    Object.keys(GAux).forEach((index) => {
        GAux[index] = {visitado: false, adj:Array()}
    })
    let u = Math.round(Math.random() * (n - 1))
    GAux[u].visitado = true
    let arestas = 0
    while( arestas < n - 1){
        let v = Math.round(Math.random() * (n -1))
        if(!GAux[v].visitado){
            GAux[v].adj.push(u)
            GAux[u].adj.push(v)
            GAux[v].visitado = true
            arestas++
        }
        u = v
    }
    return Object.keys(GAux).map(index => GAux[index].adj)
}

const eh_arvore = G => {
    const arestas = numero_arestas(G)
    if(arestas != (G.length - 1)){
        return false
    }
    const s = Math.floor(Math.random() * (G.length - 1))
    const aux = bfs(G, s)
    for(let i = 0; i < G.length; i ++){
        if(aux[i] === Number.POSITIVE_INFINITY)
            return false
```

```
    }
    return true
}

const teste_arvore = () => {
    const n = [250, 500, 750, 1000, 1250, 1500, 1750, 2000]
    n.forEach(number => {
        let soma_diametro = 0
        for(let i=0; i<500; i++) {
            let G = random_tree_random_walk(number)
            assert(eh_arvore(G))
            soma_diametro = soma_diametro + diametro(G)
        }
        let media = soma_diametro/500
        console.log(number + ' ' + media)
    })
}


teste_aresta()
teste_arvore()
```

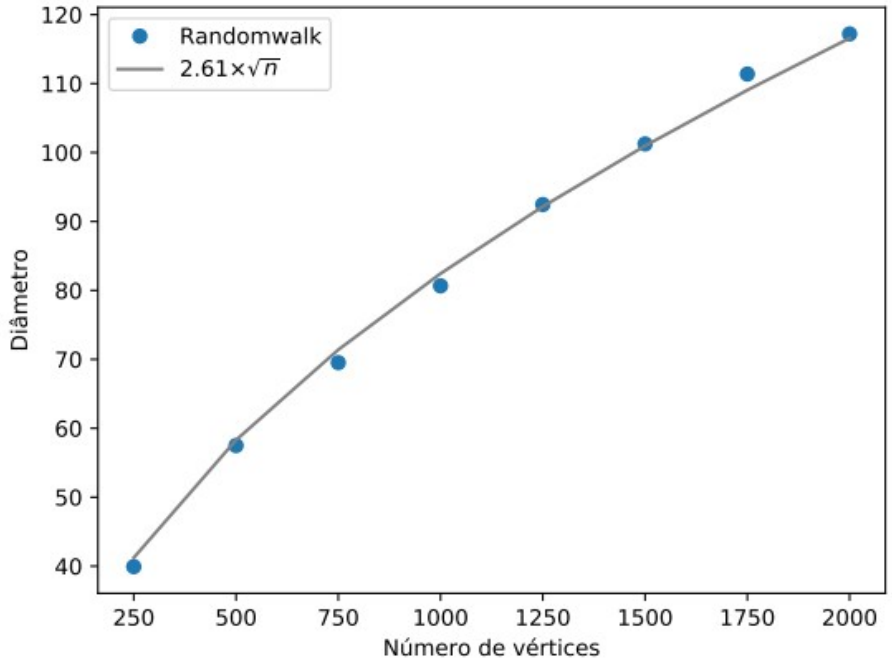

Figure 1: Diâmetro médio da árvores geradas pelo algoritmo

OBS: O código completo com as outras funções implementadas podem ser vistas no repositório do github:
https://github.com/thiagoissao/arvore_aleatorias_grafos/blob/master/diametro.js