

Prática 7

Para a prática foi utilizado o código do prf. Ziviani “7.15-7.17-dijkstra”, porém o código em questão prevê apenas um peso por aresta, na prática é necessário que seja utilizado dois pesos para a definição do melhor caminho. A primeira modificação foi na criação da matriz que foi criada da seguinte maneira

```
private int mat[][][];
```

De modo que os primeiros dois campos, são referentes aos vértices, e o terceiro campo referente ao peso e possui tamanho dois, se o valor do terceiro campo for **0**, o peso é referente à *distância* e se for **1** é referente ao *tempo*. Segue o construtor da classe.

```
private static final byte dist = 0;
private static final byte temp = 1;

public JGrafo (int numVertices) {
    this.mat = new int[numVertices][numVertices][2];
    this.pos = new int[numVertices];
    this.numVertices = numVertices;
    for (int i = 0; i < this.numVertices; i++) {
        for (int j = 0; j < this.numVertices; j++){
            this.mat[i][j][temp] = 0;
            this.mat[i][j][dist] = 0;
        }
        this.pos[i] = -1;
    }
}
```

A segunda modificação foi na construção do heap, o critério utilizado foi que o menor caminho deve ser a menor soma dos pesos, tanto *peso* quanto *tempo*. Essa modificação pode ser vista no trecho a baixo.

```
public void refaz (int esq, int dir) {
    int j = esq * 2;
    int x = this.fp[esq];
    while (j <= dir) {
        if ((j < dir) && (this.pDist[fp[j]] + this.pTemp[fp[j]] > this.pDist[fp[j + 1]] + this.pTemp[fp[j + 1]])) j++;
        if (this.pDist[x] <= this.pDist[fp[j]]) break;
        this.fp[esq] = this.fp[j]; this.pos[fp[j]] = esq;
        esq = j; j = esq * 2;
    }
    this.fp[esq] = x; this.pos[x] = esq;
}
```

Saída do exemplos da prática.

Exemplo 1

1

4

5

Exemplo 2

3

4

0

a=0; b=1, c=2, d=3, e=4