

Practica 2

Computación Distribuida 2021-1

Profesora: Karla Vargas

Diego Estrada

Jorge Cortés

Ricardo Desales

José Morán

23 de octubre de 2020

1. Introducción

En esta práctica tendrás que implementar los algoritmos DFS [Raynal, 27] y BFS que no detecta terminación [Aspnès, 31].

2. Desarrollo

Esta práctica estará conformada por tres archivos (con sus respectivas interfaces) y un test:

- `NodoBFS.py`
- `NodoDFS.py`
- `CanalRecorridos.py`
- `Test.py`

Tendrás que implementar las semánticas de los nodos de cada uno de los algoritmos que corresponden. Siguiendo la estructura que se les proporciona en cada uno de ellos.

3. Pruebas

Para la ejecución de las pruebas¹ basta con ejecutar el siguiente comando en la terminal:

```
myUser $> pytest -q test.py
```

4. Observaciones

- Respeta los constructores proporcionados.
- Por convención, el nodo distinguido en todos los algoritmos será el nodo con el índice 0.
- Por convención llamaremos `padre` y `distancia` a los atributos del nodo i que almacenarán el id del padre y la distancia al nodo distinguido en el algoritmo de BFS; a su vez en el DFS las llamaremos `padre` y `vecinos` a los atributos que necesitamos. Es importante que estos valores estén correctamente asignados a estas variables, ya que de otra manera el test no te servirá.

¹Insistimos, pasar las pruebas no implica que su práctica esté correcta. Sin embargo ayudan a mantener cierta estructura entre todas las diferentes implementaciones y son guía para que visualicen el comportamiento de sus implementaciones con respecto a resultados que se conocen como correctos.

- En BFS y DFS al crear un nodo, por convención, haremos que la referencia al padre sea él mismo (es algo no especificado en los algoritmos).
- En DFS debido a que el algoritmo tiene una naturaleza no determinista, haremos una pequeña modificación: en lugar de que para cada nodo el algoritmo en la elección de un nuevo vecino tome un vértice aleatorio, se elegirá el vértice vecino con el menor id.

Esto es si

$$N_{nv}(v_i) = \{v_3, v_6, v_2, v_7, v_9, v_8\}^2$$

entonces el vértice elegido para continuar la ejecución del algoritmo sería v_2 .

- Cualquier duda escríbanos a la brevedad.

²Donde $N_{nv}(v_i)$ es el conjunto de vértices vecinos no visitados del vértice v_i