

Computación Distribuida 2024-1

- Practica 3 BFS y DFS sin terminación-

Profesor: Fernando Michel Tavera

Ayudantes: Yael Antonio Calzada Martín y Mauricio Riva Palacio Orozco

Fecha de entrega: Martes 8 de Octubre 2024

1. Descripción Practica

En esta practica deberán implementar los algoritmos DFS Y BFS que no detectan terminación. El pseudocódigo se encuentra en la carpeta practica 3

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. Prueba (Test.py)

Estas pequeñas pruebas tienen la finalidad de validar los resultados de sus implementaciones. **Ojo:** que las pruebas unitarias sean pasadas con éxito no implica que su calificación sea la máxima. Así que tómennas como un apoyo para el diseño de sus algoritmos.

3.1. Prerrequisitos

Para ejecutar las pruebas basta con ejecutar el siguiente comando en la terminal:

```
pytest -q test.py
```

4. Observaciones

- Respetar 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.
- 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

5. Lineamientos de entrega

Dado que el esqueleto de la práctica está hecho y se espera que solamente se rellene, se debe entregar cada clase bien documentada, además dentro de la práctica se deberá agregar un ReadMe con el número de la práctica, en este caso Práctica 3, nombre de los integrantes con su número de cuenta y una explicación de su implementación para cada algoritmo.

Solo un integrante debe subir la práctica, el otro integrante con que la marque como entregada antes de la fecha límite es suficiente, además, se deberá subir la carpeta comprimida como un .zip con el nombre "Practica3"

Cualquier duda escríbanme a la brevedad.