Computación Distribuida 2023-2
- Practica 2 Algoritmos-

Profesor: Fernando Michel Tavera
Ayudantes: Yael Antonio Calzada Martín y Mauricio Riva Palacio Orozco

Fecha de entrega: Viernes 20 de Septiembre 2024

1. Descripción Practica

Para la entrega de la práctica recuerda respetar los lineamientos de entrega.
Deberán implementar las siguientes funciones en Simpy:

1. El algoritmo para conocer a los vecinos de mi vecinos ya lo he dado pues es el NodoVecinos. (Algoritmo 1 del archivo Algoritmos.pdf) de igual forma para resolverlo el HINT seria "¿Los canales cambian?"

2. Implementar el algoritmo para construir el árbol generador (Algoritmo 4)

3. Implementar el algoritmo de Broadcast (Algoritmo 5).

4. EXTRA: Implementar el algoritmo de Convergecast (Algoritmo 6).

Recuerden que para cada algoritmo deben usar una implementación de un Canal (algunos canales pueden servir para distintos algoritmos, otros requerirán su canal especial) y una de Nodos (única para cada algoritmo). Metan sus Canales a la carpeta correspondiente y dejen los nodos donde está Nodo.py.

La clase Nodo.py, NodoVecinos.py y Canal.py no es necesario/no deberían hacerles modificaciones, es el código que ya les doy yo de base

2. Prueba (Test.py)

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

2.1. Prerrequisitos

Para correr la practica primero deberán instalar python y simpy con el siguiente comando:

```
sudo apt update
```

```
sudo apt install python3
```

```
python3 --version
```

```
pip install simpy
```

En caso de que estén trabajando con un subsistema de Linux en Windows, puede que deban instalar antes pip, se instala con el siguiente comando:

```
sudo apt-get install python3-pip
```

Si el comando para instalar pip3 falla pueden intentar con:

```
sudo apt-get update
```

Para el uso de las pequeñas pruebas unitarias es necesario que se cuente con la biblioteca pytest instalada en sus equipos. Lo anterior lo pueden realizar mediante el comando siguiente comando en su terminal:

```
myUser $> pip3 install pytest$
```

3. Modificaciones a sus clases

El no darles el una estructura fija para el desarrollo de sus prácticas puede ser razón de conflictos al momento de hacer tests generalizados. Es por ello que les pedimos que realicen unas pequeñas secciones de su código. A partir de la siguiente práctica será proporcionada una estructura homologada para todos, junto con las pruebas unitarias.

3.1. Nombre de variables

- **Prueba para conocer a los vecinos de los vecinos de un nodo**
  identifiers (n.identifiers):
  Dicha variable debe corresponder con una lista donde cada nodo almacena a los otros nodos que conoce. La prueba arroja un error mostrando el nodo que tiene mal dicho valor de la variable.

- **Prueba el algoritmo que construye el árbol generador**
  padre (n.padre):
  referencia al nodo padre del nodo en cuestión hijos **(n.hijos):**referencia a una lista de nodos que son los hijos del nodo en cuestión.

- **Prueba para el algoritmo Broadcast**
  mensaje (nodo.mensaje):
  variable que hace referencia al mensaje que fue enviado por el nodo p_s

- **Prueba para el algoritmo de convergecast**
  Al ser extra, no hay prueba para este ejercicio.

4. Clases y nombres de funciones y métodos.

La estructura que les propuse fue que una gráfica la podíamos ver como una lista de vértices donde el vértice v_i se encuentra en la localidad i^3 de la lista y

5. Uso

Para el uso de las pruebas sigue los siguientes pasos:

- Localiza en la misma carpeta que tus códigos fuentes el archivo test.py
- Ejecuta el siguiente comando:

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

6. Lineamientos de entrega

Dado que el esqueleto de la practica esta 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 practica, en este caso Practica 2, nombre de los integrantes con su número de cuenta y una explicación de su implementación para cada algoritmo, en caso de que el equipo resolviera el punto extra, la clase del punto extra deberá llamarse 'NodoConvergecast.py' y en caso de que creen otro canal para el ejercicio extra deberá llamarse 'CanalConvergecast.py'. Cabe mencionar que el punto extra no viene incluido en el test.py, así que deberán incluir una explicación de como correrlo. (Más libertad para que lo implementen desde 0.)
Solo un integrante debe subir la practica, 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 "Practica2"