

Computación Distribuida

Práctica 2

Pablo Gerardo González López
pablog@ciencias.unam.mx

Daniela Susana Vega Monroy
danisu@ciencias.unam.mx

Luis Fernando Yang Fong Baeza
fernandofong@ciencias.unam.mx

Diego Estrada Mejia
diegoe@ciencias.unam.mx

Miguel Ángel Piña Avelino
miguel_pinia@ciencias.unam.mx

23 de septiembre de 2021

Para esta práctica se cuentan con dos scripts, el script `Graph.ex` y `Tree.ex`. Ambos scripts tienen el módulo de una Gráfica y de un Árbol que ya generan un número n de nodos activos con la función `loop` en estructura de gráfica y árbol respectivamente, el propósito de esta práctica es completar estas funciones para que se ejecuten de manera correcta los algoritmos de broadcast, convergecast, BFS y DFS en su versión distribuida.

1. Árboles distribuidos

Para este script, el script `Tree.ex`, ya se genera un árbol binario completo de tamaño n , cumpliendo con la propiedad de que para cualquier nodo i en el árbol, sus hijos se encontrarán en las posiciones $2i + 1$ para el hijo izquierdo y $2i + 2$ para el hijo derecho en caso de que dichos índices sean menores que n , es decir, que existan.

Para que ambas funciones sean completadas de manera exitosa, se debe de cumplir la siguiente restricción: *Al final del algoritmo se le debe de los nodos finales al nodo principal para informar del terminamiento del mismo*, es decir, para el caso de Convergecast, el nodo raíz debe de enviar un mensaje al nodo principal informando que el algoritmo fue ejecutado exitosamente, en el caso de un Broadcast, el nodo principal debe de recibir k mensajes con k el número de hojas en el árbol informando que cada hoja recibió el mensaje de que un broadcast fue ejecutado. **Hint:** Si un árbol tiene n

nodos, entonces tiene $\lfloor \frac{2n+1}{2} \rfloor$ hojas.

Para la función broadcast, basta con regresar una tupla de la forma `{raiz, status}`, en el caso de la función broadcast, se debe de regresar una lista de k tuplas de la misma forma que la función broadcast, una por cada hoja.

2. Gráficas distribuidas

Para este caso, cada nodo sabe su distancia actual, que está representada por la variable de estado, el propósito de estos scripts es ejecutar mediante un paso de mensajes los algoritmos de BFS y DFS en su versión distribuida

Hint: Tener en cuenta que los nodos siempre van a estar recibiendo mensajes, ya sea del nodo principal o de cualquier nodo vecino, sin embargo, existe un punto en el que la gráfica se tiene que establecer para ambos algoritmos, es decir, no importa cuántos mensajes se envíen, la gráfica en sí converge a ciertas distancias para cada nodo dependiendo el vértice origen, por esto, se debe de incluir en algún momento la línea `Process.sleep(5000)` que hace que un hilo se duerma 5 segundos.

Para este script existe un punto extra, para el equipo que implemente ambos algoritmos con una sola función, en caso de obtener 11 de calificación, se les abonará a cualquier otra práctica que se vea en el curso.

3. Fecha de entrega

La entrega es antes del día 13 de Octubre a las 23:59:59 vía Classroom.