

Computación Distribuida

Tarea 2

González Montiel Luis Fernando

Rubí Rojas Tania Michelle

27 de agosto de 2019

1. En la Ciudad de México varios puntos de control (nodos) fueron agregados. Cada uno de estos puntos tiene información de cuantas personas viven dentro de cierto diámetro. Un punto (raiz) de la delegación Benito Juárez quiere recolectar la información de todos los puntos de control y determinar cuántas personas viven en la CDMX. Escribe un algoritmo distribuido para que cada nodo recolecte la información de sus hijos para enviársela al padre sin repetirla y que el padre reporte el total de pobladores viviendo la CDMX.

SOLUCIÓN:

```
initially do{
    parent = null;
    children = null;
    visited = null;
    if(Pi = Ps){
        parent = Pi
        let Pk \in neighborsi
        * send # personas
        send GO to Pk
    }
    upon the receiving of GO from Pj
    if(parent = null){
        * parent = Pj;
        * visited = {Pj} && {#personas}
        if(visited=neighbors){
            send BACK(yes) to Pj;
        }
    }*else{
        let Pk \in neighbors visited;
        send GO to Pk;
    }else{
        send BACK(no) to Pj;
    }
    upon the receiving of BACK(resp)
    if(resp=yes){
        * children = children \cup {Pj}
        * visited = visited \cup {Pj} && {#personas}
        if(visited=neighbors){
            if(parent = Pj){
                * return sumaTotal#personas
            }*else{
                send BACK(yes) to parent
            }
        }else{
            let Pk \in neighbors \ visited
            send GO to Pk
        }
    }
}
```

```

    }
  }
}

```

2. Explica por qué el algoritmo de convergecast con broadcast puede producir más de un árbol.

SOLUCIÓN:

Porque como no tiene retardos en las aristas,
es indiferente a que vértice se escoja para visitar,
entonces eso te daría varios posibles árboles generados.

3. Para el algoritmo BFS visto en clase, si las aristas tienen peso indicando que es el tiempo que toma en llegar un mensaje del vértice p al q , ¿Cuánto tiempo le toma al algoritmo BFS formar el árbol?

SOLUCIÓN:

Le toma tiempo $O(n^2)$ porque para $|e| = (n(n-1)) / 2$
y esto quitando las constantes quedaría como $(n)(n) = n^2$.

4. Explicar por qué el algoritmo Awerbuch DFS tiene complejidad de tiempo $O(V)$
tiempo (tarda en total $4|V| - 2$) y $O(E)$ mensajes (envía $4|E|$ mensajes)

SOLUCIÓN:

Porque encuentra un árbol DFS en una gráfica arbitraria en $O(n)$ tiempo y $3(|E|)$.
Una arista del árbol se recorre 2 veces, 2 tokens en cada dirección
y 1 en vis ya que solo envía pero no cuenta como recorrido.