

Suponga una ciudad representada por una matriz $A(n \times n)$. De cada esquina x, y se conocen dos valores enteros que representan la cantidad de autos y motos que cruzaron en la última hora. Los valores de cada esquina son mantenidos por un proceso distinto $P(x, y)$. Cada proceso puede comunicarse sólo con sus vecinos izquierdo, derecho, arriba y abajo, y también con los de las 4 diagonales (los procesos de las esquinas tienen sólo 3 vecinos y los otros en los bordes de la grilla tienen 5 vecinos).

Escriba un algoritmo heartbeat que calcule **las esquinas** donde cruzaron la **mayor cantidad de autos** y la **menor cantidad de motos** respectivamente, de forma que al terminar el programa cada proceso conozca ambos valores.

```
chan topologia[1:n] ([1:n,1:n] bool)
Process Nodo[p:1..n]{
    bool vecinos[1:n]                // Arreglo q en la posición N tiene true si N es vecino
    int max = autos, min = motos      // Asumimos que estas vienen inicializadas
    int esqMax, esqMin = p;
    int nuevoMax, nuevoMin, nuevaesqMax, nuevaesqMin;

    for (i < n-1){                    // FOR que se ejecuta el mínimo de veces necesarias para que
                                        // lleguen los datos entre los dos nodos más lejanos
        for [q = 1 to n st vecinos[q] ] // Le manda un min y max a sus vecinos a sus canales
            send topologia[q](max, min, esqMax, esqMin); // Notar que usa PMA. Con PMS es más difícil coordinar.

        for [q = 1 to n st vecinos[q] ] { // Recibe min y max de cada vecino por su canal
            receive topologia[p](nuevoMax, nuevoMin, nuevaesqMax, nuevaesqMin)
            if(maximo < nuevoMax)          // actualiza maximos locales si es necesario
                maximo = nuevoMaximo;
                esqMax = nuevaesqMax;

            if(minimo > nuevoMinimo)        // Actualiza mínimos locales si es necesario
                minimo = nuevoMinimo;
                esqMin = nuevaesqMin;
        }
    }
}
```

Nota: Usar mecanismo de pasaje de mensajes justificando la elección del mismo

Se utiliza PMA porque es más fácil coordinar los ciclos de envío y recepción de mensajes a los vecinos que con PMS.

b) Analice la solución desde el punto de vista del número de mensajes.

Para una matriz de tamaño $N \times N$ con $N > 1$

Nota: no se si están bien hechas las cuentas. Tampoco se si hace falta hacer las cuentas, por ahí con decir que es de hay $N \times N$ nodos que se ejecutan $N-1$ rondas alcanza para justificar que la cant de mensajes es de Orden N^3

En una ronda:

Mensajes no enviados por los nodos de los bordes:

Cada nodo de borde envía 5 en lugar de 8: no envía 3

Cada borde no envía $N \times 3$

Hay 4 bordes: no se envían $(N \times 3) \times 4$

Pero las esquinas las restamos 2 veces, así que las sumo de nuevo: $(N \times 3) \times 4 + 4$

Cantidad de mensajes:

$(N \times N \times 8) - \text{MensajesNoEnviados}$

$= (N \times N \times 8) - ((N \times 3) \times 4 + 4)$

$= 8 N^2 - 12N - 4$

En todas las rondas:

Cantidad de rondas: $N-1$

Cantidad de mensajes:

$(8 N^2 - 12N - 4) \times (N-1)$

$= (8N^3 - 12 N^2 - 4N) - (8 N^2 - 12N - 4)$

$= 8N^3 - 12 N^2 - 4N - 8 N^2 + 12N + 4$

$= 8N^3 - 20 N^2 + 8N + 4$

Cantidad de mensajes total:

$8N^3 - 20 N^2 + 8N + 4$

La cantidad de mensajes es de Orden N^3

c) Analice y describa si puede realizar alguna mejora para reducir nro de msjs

Se puede reducir la cantidad de mensajes si los nodos terminan de trabajar apenas conocen el máximo y mínimo global. Los nodos centrales de la matriz conocen estos valores antes que los de los bordes porque la distancia a su nodo más lejano es menor. Al terminar antes se ejecutan menos rondas y por lo tanto envían menos mensajes. Ejemplo: el nodo central de una matriz de 5x5 sólo necesita 2 rondas para conocer el máximo y mínimo global y puede detenerse en ese momento, mientras que los de las esquinas se siguen ejecutando porque necesitan 4 rondas para terminar su ejecución.

d) Qué modificaría de la solución original si no existiera diagonales

Si no hubiera diagonales los se reduciría la cantidad de mensajes por ronda ya que cada nodo tiene a lo sumo 4 vecinos en lugar de 8. Pero al mismo tiempo la cantidad de rondas aumentaría porque la distancia entre los 2 nodos más lejanos aumenta.

En lugar de haber $N-1$ rondas va a haber $2*(N-1)$