

Final Concurrente 22/02/17

1. Dado el siguiente bloque de código, indique para cada uno de los items si son equivalentes o no. Justificar cada caso (de ser necesario, de ejemplos).

SEGMENTO 1	SEGMENTO 2
<pre>... int cant = 1000; while(true) { if (cant > 15); datos?(cant)→Sentencias1 □ (cant < 5); datos?(cant) →Sentencias2 □ (<u>INCOGNITA</u>); datos?(cant)→Sentencias3 end if } ...</pre>	<pre>... int cant = 1000; do (cant > 15); datos?(cant)→Sentencias1 □ (cant < 5); datos?(cant) →Sentencias2 □ (<u>INCOGNITA</u>); datos?(cant)→Sentencias3 end do ...</pre>

- a) INCOGNITA equivale a: ($cant = 5$) or ($cant = 15$)
b) INCOGNITA equivale a: ($cant > 0$)
c) INCOGNITA equivale a: ($cant \geq 2$) and ($cant \leq 20$)
d) INCOGNITA equivale a: ($cant > 5$) or ($cant \leq 15$)
e) INCOGNITA equivale a: ($cant > 5$) or ($cant < 15$)

2. Suponga una ciudad representada por una matriz A(nxn). 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).

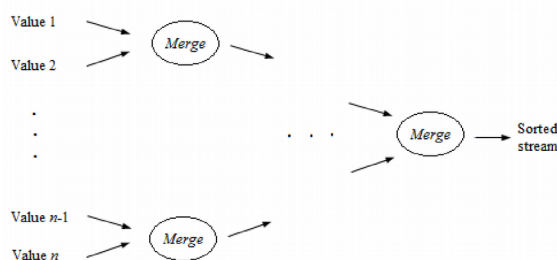
- a) Escriba un algoritmo heartbeat que calcule las esquinas donde cruzaron la mayor cantidad de autos y la menor cantidad de motos respectivamente, de forma que al terminar el programa cada proceso conozca ambos valores.
b) Analice la solución desde el punto de vista del número de mensajes.
c) Analice y describa si puede realizar alguna mejora para reducir el número de mensajes.
d) Qué se modificaría de la solución original si no existieran las diagonales?
NOTA: utilice un mecanismo de pasaje de mensajes, justificando la elección del mismo.

3. Describa los mecanismos de RPC y Rendezvous. Detalle características, similitudes, diferencias, ventajas y desventajas.

4. a) Cuál es el objetivo de la programación paralela?
b) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup (S) está regido por la función $S=p-10$ y en el otro por la función $S=p/3$ (para $p>10$). ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores?
c) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10.000 unidades de tiempo, de las cuales sólo el 90% corresponde a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo?

5. Suponga los siguientes métodos de ordenación de menor a mayor para n valores (n par y potencia de 2), utilizando pasaje de mensajes:

- i- Un pipe de filtros. El primero hace input de los valores de a uno por vez, mantiene el mínimo y pasa los otros al siguiente. Cada filtro hace lo mismo: recibe un stream de valores desde el predecesor, mantiene el más chico y pasa los otros al sucesor.
ii- Una red de procesos filtro (como la de la figura).
iii- Odd/even exchange sort. Hay n procesos $P[1:n]$. Cada uno ejecuta una serie de rondas. En las rondas “impares”, los procesos con número impar $P[\text{impar}]$ intercambian valores con $P[\text{impar}+1]$. En las rondas “pares”, los procesos con número par $P[\text{par}]$ intercambian valores con $P[\text{par}+1]$ ($P[1]$ y $P[n]$ no hacen nada en las rondas “pares”). En cada caso, si los números están desordenados actualizan su valor con el recibido.



Nota: cada proceso tiene almacenamiento local sólo para 2 valores (el próximo y el mantenido hasta ese momento).

- a) Cuántos procesos son necesarios en i e ii? Justifique.
b) Cuántos mensajes envía cada algoritmo para ordenar los valores? Justifique.
c) En cada caso, ¿cuáles mensajes pueden ser enviados en paralelo (asumiendo que existe el hardware apropiado) y cuáles son enviados secuencialmente? Justifique.
d) Cuál es el tiempo total de ejecución de cada algoritmo? Asuma que cada operación de comparación o de envío de mensaje toma una unidad de tiempo. Justifique.