**Pontificia Universidad Católica Madre y Maestra**

**Campus Santo Tomás de Aquino**

**Facultad de Ciencias de la Ingeniería**



**Departamento de ISC**

**Programa:**

Programación 2

**Profesor:**

Manuel Peralta

**Tema**:

Informe 2

**Grupo:**

SD-ISC-307-T-001

**Integrantes:**

   2016-5536 - Raúl Herrera Rojas

**Día de entrega:**

26/11/2018

10.1

10.4

P0

While (1)

Sort(array)

Send (array[max])

Recv (recvValue)

If(arr[max] < recvValue)

array[max] = recvValue

else

break

P1

While (1)

Sort(array)

Recv (recvValue)

Send (array[min])

If(arr[min] > recvValue)

array[min] = recvValue

else

break

10.5

La combinacion de send y recv que están dentro del else pueden bloquear el programa, para que este funciona tenemos que invertir la posición de estos, poner el recv antes que el send y el send luego del recv. El proceso 0 puede internar llamar un proceso de la izquierda inexistente y el ultimo proceso puede intentar acceder a un proceso inexistente a la derecha.

10.6

En el proceso de los impares se cambia la condición “n – 3” por” n – 3 + n % 2” y en los procesos pares se agregan la condición para la primera combinación de recv y send “i ¡= n - 1 “

10.12

12,2,11,4,9,1,10,15,5,7,14,3,8,13,6,16

2,12,4,11,1,9,10,15,5,7,3,14,8,13,6,16

2,4,11,12,1,9,10,15,3,5,7,14,6,8,13,16

1,2,4,9,10,11,12,15,3,5,6,7,8,13,14,16

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

10.14

El algoritmo odd-even merging, puede ser aplicado para listas de diferentes tamaños. Ambas sublistas originales se pueden crear por el mismo algoritmo. El algoritmo también puede ser usado recursivamente lo cual nos lleva a una complejidad O(log2n) con una cantidad n de procesos. En este caso, los procesadores, en pares, realizan las operaciones de compare and exchange. Este algoritmo puede ser implementado con unidades de hardware que realicen operaciones de compare and exchange como fue previsto por Batcher.

Al igual que en el odd-even, el bitonic logra una complejidad de O(log2n) con n procesos (un proceso por cada número). La data puede ser particionada, como en todos los algoritmos de ordenamiento, para reducir el numero de procesadores con un aumento de asistencia en el número de pasos internos. El bitonic puede ser mapeado a una malla al igual que a un hipercubo.

El bitonic, al igual que el odd-even, es interesante para ser implementado con unidades de hardware.