

Taller 2: MIPS

Cajal Matías, Levin Nicolas, Zuñigas Cesar

Universidad de Río Negro

Arquitectura de Computadoras I

28 de abril de 2023

Indice

| | |
|---|---|
| 1) Algoritmo de ordenamiento | 3 |
| 1.a) Comparación entre implementaciones | 3 |
| 1.a.a) Caso de prueba dado por el Taller | 3 |
| 1.a.b) Peor caso de prueba | 3 |
| 1.a.c) Cantidad de instrucciones, distintos largos de arreglo | 4 |
| 2) Recursividad y stack | 5 |

1) Algoritmo de ordenamiento

1.a) Comparación entre implementaciones

1.a.a) Caso de prueba dado por el Taller

El caso de prueba dado por el taller es:

arr: 12, 255, 200, 150, 50, 33, 133, 100, 17, 15, 33, 188, 0, 10, 12, 40, 201, 7, 9, 10

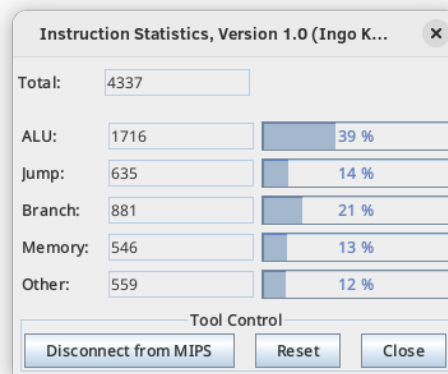


Figura 1: “Cantidad de instrucciones usando la función del punto a.”

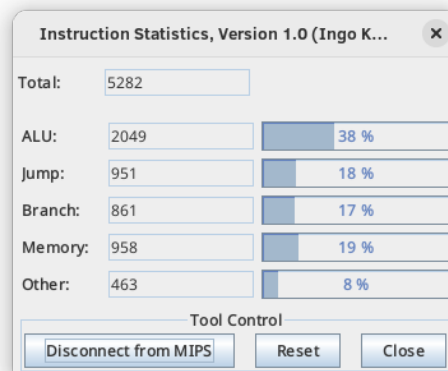


Figura 2: “Cantidad de instrucciones implementando ordenamiento burbuja.”

El método burbuja es más ineficiente porque debe intercambiar de a una posición del arreglo. Por este mismo motivo, utiliza muchas más instrucciones tanto de tipo R como de tipo I.

Para este caso de prueba el método burbuja utilizó un **19,4%** más de instrucciones de la ALU, **49,76%** extra en saltos y un **75,09%** de más en operaciones de memoria.

1.a.b) Peor caso de prueba

El peor caso es un arreglo que este ordenado de menor a mayor, ya que debe ser invertido en su totalidad:

arr: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

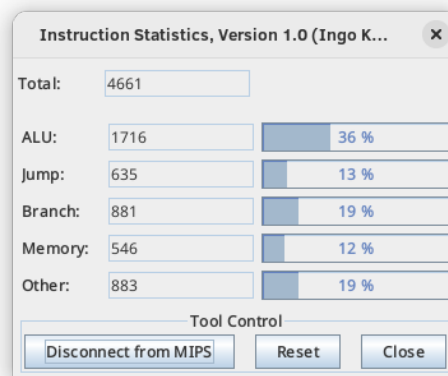


Figura 3: “Cantidad de instrucciones usando la función del punto a.”

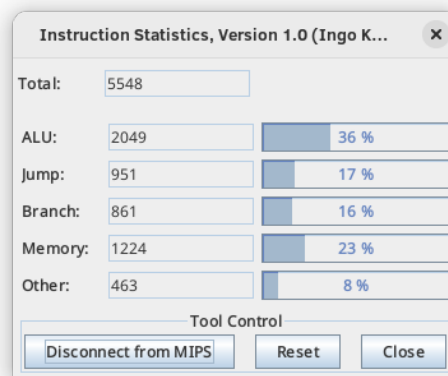


Figura 4: “Cantidad de instrucciones implementando ordenamiento burbuja.”

Para el ordenamiento burbuja se utilizaron un **19,03%** más de instrucciones.

1.a.c) Cantidad de instrucciones, distintos largos de arreglo

En la siguiente tabla se comparan las dos implementaciones en base a la cantidad de elementos en el arreglo. Operando con el peor caso.

| Cantidad de elementos | Ejercicio B | Metodo Burbuja |
|-----------------------|--------------|----------------|
| 10 | 1641 | 1590 |
| 20 | 4461 | 5548 |
| 30 | 9661 | 11898 |
| 40 | 16461 | 20648 |
| 100 | 71642 | 110589 |
| 1000 | 3291496 | 7776692 |
| 1000000 | 270913392133 | 2709217949917 |

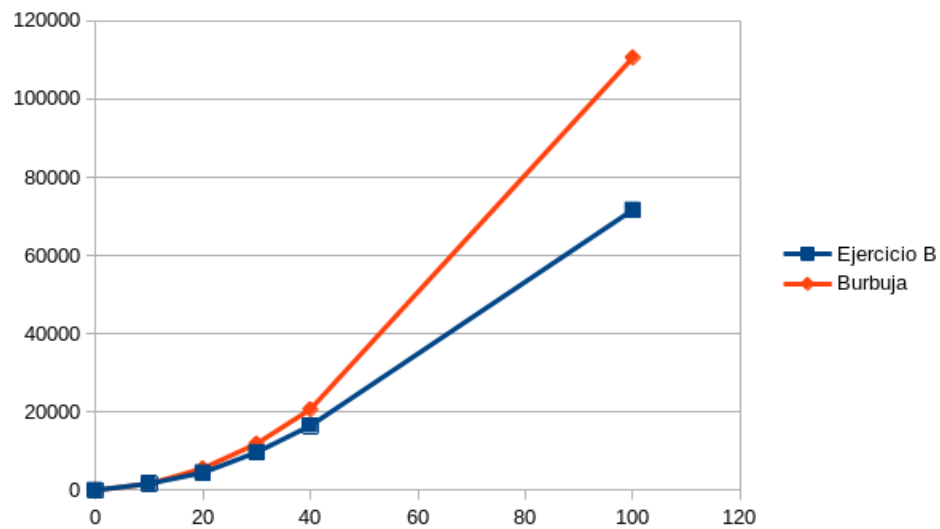


Figura 5: “Grafico comparando desde arreglo con 10 elementos hasta 100”

La cantidad de instrucciones crece de manera exponencial en relación a la cantidad de elementos dentro del arreglo.

2) Recursividad y stack

Se puede calcular hasta el factorial de 12 inclusive, en adelante genera overflow. Esto se debe a una limitación de la arquitectura, los registros pueden operar hasta 32 bits. El factorial de 12 es el último factorial capaz de representarse en 32 bits.