Benchmark: “Remove Duplicates”

Author: Issa Qandah *Reviewed by: “Hassan TaqiEddin”*

Description & Notes

- This Benchmark removes duplicates from an array by iterating through each element.
- And make a new array that contains only unique elements.
- Time Complexity O(N^2).

Algorithm (Pseudo or C)

```
Initialize:
  array = [5, 1, 3, 5, 1, 2, 3, 4, 5, 6, 7, 6, 0]
  result = []

For each element in array:
  If element is not in result:
     Add element to result

Output result
```

Registers and memory used in implementation

$8 : size of the array

$9 : outer loop index (i)

$10 : write index for result array

$11 : temporary register for duplicate check

$14 : inner loop index (j)

$15 : flag for duplicate detection (0: no, 1: yes)

$13 : current element array[i]

$16 : current result[j]

$25 : address of result[j]

$12 : address of array[i]

$24 : tmp reg to compare inner loop with write index

Code (.data and .text)

```
.data
array:     .word 5, 1, 3, 5, 1, 2, 3, 4, 5, 6, 7, 6, 0, 10
result:    .space                                # Space for the resulting array

.text
    ADDI $8 , $0, 14                         # $8 = size of array (n)
    ADDI $9, $0, 0                           # $9 = index i (outer loop index)
    ADDI $10, $0, 0                          # $10 = write index for result array

outerLoop:
    SLT $11, $9, $8                          # $11 = 1 if $9 < $8
    BEQ $11, $0, finish                     # Exit loop if $9 >= $8

    # Choose one of these Insertion based on your memory
    # For Word addressable       # For byte addressable
    # ADD $12, $9, $0              SLL $12, $9, 2
    LW $13, array($12)                     # $13 = array[i]
    ADDI $14, $0, 0                        # $14 = inner loop index j
    ADDI $15, $0, 0                        # $15 = flag (0: no duplicate, 1: duplicate)

innerLoop:
    SLT $24, $14, $10                       # $24 = 1 if $14 < $10
    BEQ $24, $0, addToResult

    # Choose one of these Insertion based on your memory
    # For Word addressable            # For byte addressable
    # ADD $25, $14, $0              SLL $25, $14, 2

    LW $16, result($25)     # $16 = result[j]
    BEQ $13, $16, duplicateFound
    ADDI $14, $14, 1
    JAL innerLoop

duplicateFound:
    ADDI $15, $0, 1                         # Mark as duplicate
    JAL outerContinue

addToResult:
    BEQ $15, $0, storeValue            # Only store if not a duplicate
    JAL outerContinue

storeValue:
    # Choose one of these Insertion based on your memory
    # For Word addressable       # For byte addressable
    # ADD $25, $10, $0              SLL $25, $10, 2

    SW $13, result($25)                  # Store array[i] in result array
    ADDI $10, $10, 1                      # Increment write index

outerContinue:
    # Increment outer loop index i
    ADDI $9, $9, 1
    JAL outerLoop

finish:
```

Expected Output

```
Memory[13…21]  = 5, 1, 3,2,4,6, 7, 0, 10
```