

# 7

# MIPS Functions and the Stack Segment

## 7.6 In-Lab Tasks

1. The function `merge_sort` sorts array **A** recursively with the help of a second `merge` function. Translate these two functions into MIPS code. Write a main function to allocate an array of **n** elements dynamically, read the array, call `merge_sort` and then print the sorted array.

```
void merge_sort(int A[], int n) {
    if (n > 1) {
        merge_sort(&A[0], n/2);           // Sort the first half
        merge_sort(&A[n/2], n - n/2);    // Sort the second half
        merge(&A[0], n);                 // Merge the two halves
    }
}

// merge the first n/2 elements with the last (n - n/2) elements
// merge is a leaf function that makes no function calls
void merge(int A[], int n) {
    int B[n];                             // Allocated on the stack
    int i = 0, j = n/2, k = 0;
    while (i < n/2 && j < n) {
        if (A[i] <= A[j]) { B[k++] = A[i++]; }
        else { B[k++] = A[j++]; }
    }

    while (i < n/2) { B[k++] = A[i++]; }
    while (j < n) { B[k++] = A[j++]; }

    for (k=0; k < n; k++) { A[k] = B[k]; }
}
```