Problem Set 3 Exercise #19: Merge Lists

Reference: Lecture 8 notes

Learning objective: Sorting

Estimated completion time: 50 minutes

Problem statement:

Merge Sort is an advanced sorting technique (which is covered in CS1020). Here, we are not going to explain how Merge Sort works, but focus on one idea employed in Merge Sort, i.e. merging two sorted lists into a bigger sorted list.

For instance, given two sorted lists {-3, 8, 65, 100, 207} and {-10, 20, 30, 40, 65, 80, 90}, the merged list would be {-10, -3, 8, 20, 30, 40, 65, 65, 80, 90, 100, 207}.

Write a program **merge_lists.c** to read two sorted lists of integers (at most 50 each) and merge them in non-descending order. Your program should contain the following function to read '**size**' elements into **list**:

```
void read_list(int list[], int size)
```

and another function:

```
void merge_lists(int list1[], int size1, int list2[],
                 int size2, int list3[])
```

to merge **list1** and **list2** into **list3** such that elements in **list3** are still arranged in non-descending order.

The challenge is to avoid using nested loop in this question.

Sample run #1:

```
Number of elements in list1: 3
Enter 3 elements: 1 3 5
Number of elements in list2: 3
Enter 3 elements: 2 4 6
Merged list: [1, 2, 3, 4, 5, 6]
```

Sample run #2:

```
Number of elements in list1: 5
Enter 5 elements: -3 8 65 100 207
Number of elements in list2: 7
Enter 7 elements: -10 20 30 40 65 80 90
Merged list: [-10, -3, 8, 20, 30, 40, 65, 65, 80, 90, 100, 207]
```