CS29003 Algorithms Laboratory
Assignment 1: Sorting, Recursions and Big O notation

General instruction to be followed strictly

1. Do not use any global variable unless you are explicitly instructed so.
2. Do not use Standard Template Library (STL) of C++.
3. Use proper indentation in your code and comment.
4. **Name your file as `<roll_no>_<assignment_no>`. For example, if your roll number is 14CS10001 and you are submitting assignment 3, then name your file as `14CS10001_3.c` or `14CS10001_3.cpp` as applicable.**
5. **Write your name, roll number, and assignment number at the beginning of your program.**
6. Make your program as efficient as possible. Follow best practices of programming.
7. Submit your program on Moodle before deadline. Submissions by email or any other means will NOT be considered for evaluation.

In this assignment you generate arrays of numbers based on known functions and then sort each array in the fastest way possible.

First, we define a new sorting technique. The $f(\cdot)$-Combined Sorting technique is as follows:

▷ If the size n of the input array C is more than $f(n)$ then C is broken into 2 halves $C[0 \dots \frac{n}{2} - 1]$ and $C[\frac{n}{2} \dots n-1]$. Each of $C[0 \dots \frac{n}{2} - 1]$ and $C[\frac{n}{2} \dots n-1]$ are recursively sorted using $f(\cdot)$-Combined Sorting. Finally, the sorted subarrays $C[0 \dots \frac{n}{2} - 1]$ and $C[\frac{n}{2} \dots n-1]$ are merged by a linear time subroutine.

▷ If $n \leqslant f(n)$ then Bubble Sort is used to sort the array C.

Next, consider the two recursive functions T_1 and T_2:

$T_1(1) = 2$, $T_1(2) = 1$
$T_1(n) = T_1(n-1) - 2T_1(n-2)$ for $n > 2$
$T_2(1) = 2$, $T_2(2) = 1$
$T_2(n) = T_2(n-1) - T_2(n-2)$ for $n > 2$

These are the steps of your program:

1. Take n as input from the user.
2. Populate Array A by running function T_1 for inputs $\{1, 2, \dots, n\}$. Array B should be populated by running function T_2 for inputs $\{1, \dots, n\}$.
3. Select an $f(\cdot)$ such that the running time of $f(\cdot)$-Combined sort on n elements is $O(n \log n)$.
4. Sort Array A with $f(\cdot)$-Combined sort technique, for a proper selection of $f(\cdot)$.

5. Sort Array B with the sorting technique amongst Bubble sort, Insertion Sort, Merge sort, Bucket sort, and f(·)-Combined Sort (for some appropriate f(·)) that will have the fastest worst case running time.

6. Output the arrays A and B before and after sorting.

Submit a single .c or .cpp file. Your code should get compiled properly by gcc or g++ compiler.

Sample Output

```
Input a value for n: 6
The unsorted array A: 2 1 -3 -5 1 11
The sorted array A: -5 -3 1 1 2 11
The unsorted array B: 2 1 -1 -2 -1 1
The sorted array B: -2 -1 -1 1 1 2
```

Policy on Plagiarism

Academic integrity is expected from all the students. You should work on the assignment/exam consulting only the material we share with you. You are required to properly mention/cite anything else you look at. Any student submitting plagiarized code will be penalized heavily. Repeated violators of our policy will be deregistered from the course. Read this to know what is plagiarism.