

# Lab 3: Cocktail Sort

## 1. Objective

Write an assembly programs to implement Cocktail Sort, a bidirectional variation of the Bubble Sort algorithm.

## 2. Background

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. It traverses elements from left and moves the largest element to its correct position in the first iteration and second-largest in the second iteration and so on.

<https://www.geeksforgeeks.org/dsa/bubble-sort-algorithm/>

Sorting Algorithms Redux 03: Bubble Sort

<https://www.youtube.com/watch?v=jAoBsroEow4>

Cocktail Sort is a variation of Bubble Sort. Cocktail Sort traverses through a given array in both directions alternatively.

<https://www.geeksforgeeks.org/dsa/cocktail-sort/>

Sorting Algorithms Redux 04: Cocktail Sort

[https://www.youtube.com/watch?v=Xmx\\_6YRBaq8](https://www.youtube.com/watch?v=Xmx_6YRBaq8)

The following table compares these two algorithms:

Feature	Basic Bubble Sort	Cocktail Sort
Sorting direction	Forward only	Forward + backward passes
Loop boundaries	Stays constant	Shifts left and right boundaries
Swap flag for early exit	Yes	Yes

## 3. Lab Steps

You are given an assembly program [bubblesort.s](#) that implements the Bubble Sort algorithm as the starting point, and a main.c file as a test harness. Your task is to implement the Cocktail Sort algorithm based on the C program at <https://www.geeksforgeeks.org/dsa/cocktail-sort/>, by adding a Backward pass (right to left) in addition to the forward pass (left to right) of bubblesort. ~~Your implementation must be derived from the provided Bubble Sort algorithm, with the same code for Forward pass (left to right), and adding a separate Backward pass (right to left), i.e., you cannot write an implementation that is completely different from the provided code.~~ (This requirement is removed.)

Compile and run:

```
arm-linux-gnueabi-hf-as -o bubblesort.o bubblesort.s
```

```
arm-linux-gnueabi-hf-gcc -o main main.c bubblesort.o
```

## 4. Report

Put your implementation as a subroutine `cocktail_sort` in `bubblesort.s` and upload both this file and `main.c` file. Copy and paste your implemented Cocktail Sort algorithm into the report, and provide a line-by-line explanation of the new code that you wrote. Use the project report template and submit the report in PDF format.

### **Grading fabric:**

Correct implementation of Cocktail Sort: 5 pts; line-by-line explanation of part (1): 4 pts

Use of project report template: 1 pt