

BME1901 – Introductory Computer Sciences

Laboratory Handout – 7

OBJECTIVES

Learn about,

- Bubble sort algorithm
- Built-in sort() function
- tic toc time keeping functions
- Solving various questions

TOOLS

Bubble sort algorithm^{1, 2, 3}

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. In each pass only a single value is put in the correct place. The algorithm, which is a comparison sort, is named for the way smaller or larger elements “bubble” to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems. Therefore this algorithm is not suitable for large data sets.

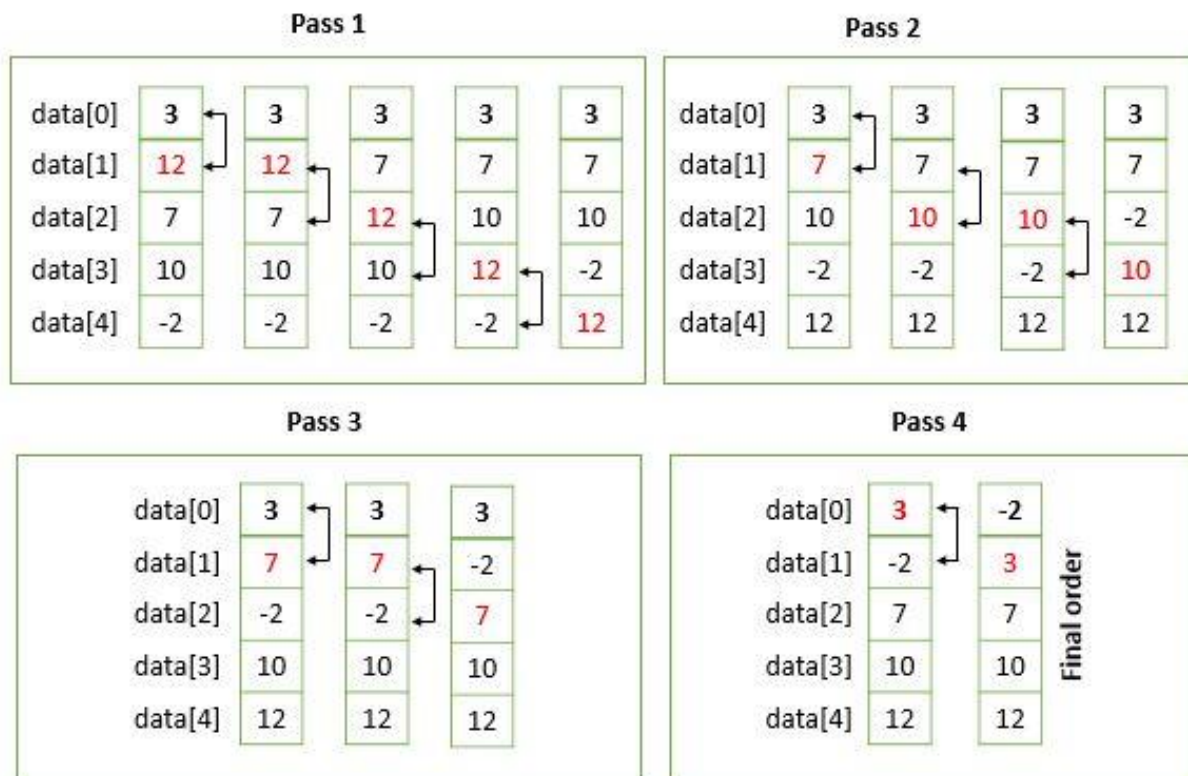


Figure 1: How bubble sort works.⁴

¹ https://en.wikipedia.org/wiki/Bubble_sort

² <https://www.cs.cmu.edu/~adamchik/15-121/lectures/Sorting%20Algorithms/sorting.html>

³ https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

⁴ <http://www.trytoprogram.com/images/working-bubble-sort.jpg>

BME1901 – Introductory Computer Sciences
Laboratory Handout – 7

Built-in sort() function⁵

Built-in sort(A) function in MATLAB sorts the elements of the input “A” in ascending order. If A is a vector, then sort(A) sorts the vector elements. If A is a matrix, then sort(A) treats the columns of A as vectors and sorts each column. Using sort(A ,direction) returns sorted elements of A in the order specified by direction using any of the previous syntaxes. 'ascend' indicates ascending order (the default) and 'descend' indicates descending order.

```
>> A = [9 0 -7 5 3 8 -10 4 2];  
  
>> B = sort(A)  
  
B =  
    -10     -7      0      2      3      4      5      8      9
```

```
>> A = [9 0 -7 5 3 8 -10 4 2];  
  
>> B = sort(A,'descend')  
  
B =  
      9      8      5      4      3      2      0     -7    -10
```

```
>> A = [10 -12 4 8; 6 -9 8 0; 2 3 11 -2; 1 1 9 3]  
  
A =  
    10    -12      4      8  
      6     -9      8      0  
      2      3     11     -2  
      1      1      9      3  
  
>> B = sort(A)  
  
B =  
      1    -12      4     -2  
      2     -9      8      0  
      6      1      9      3  
     10      3     11      8
```

⁵ <https://www.mathworks.com/help/matlab/ref/sort.html>

BME1901 – Introductory Computer Sciences

Laboratory Handout – 7

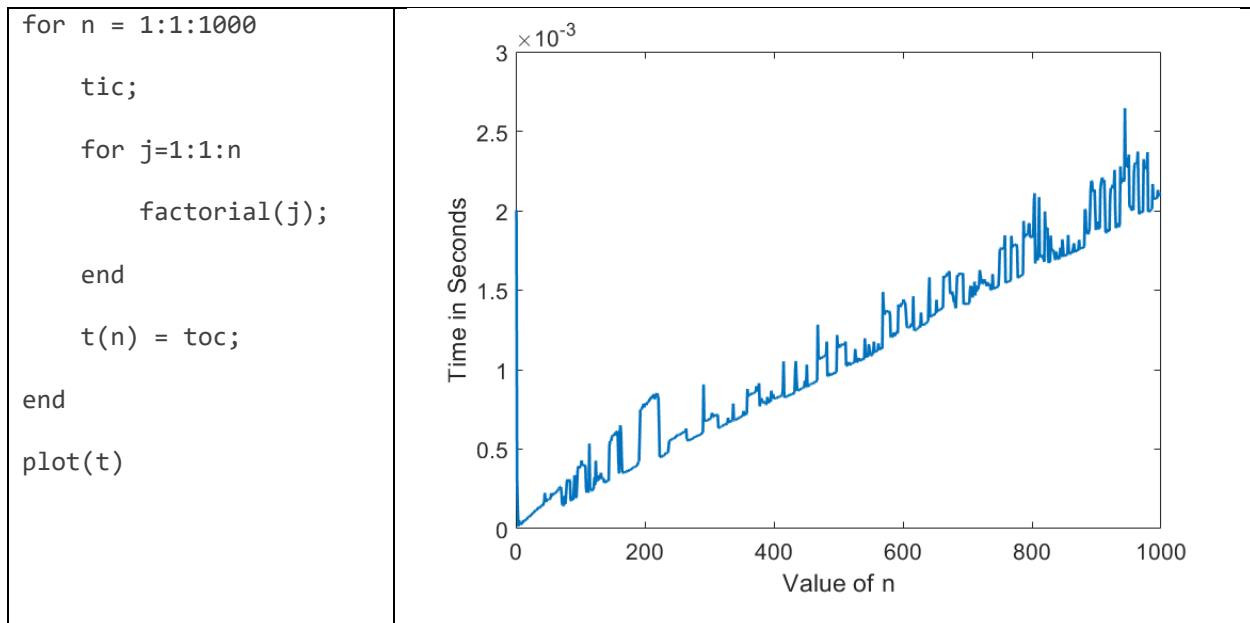
tic toc time keeping functions^{6,7}

“tic” starts a stopwatch timer to measure performance. The function records the internal time at execution of the “tic” command. Display the elapsed time with the “toc” function.

“toc” reads the elapsed time from the stopwatch timer started by the “tic” function. The function reads the internal time at the execution of the “toc” command, and displays the elapsed time since the most recent call to the “tic” function that had no output, in seconds.

tic	tic
statements	statements
toc	t = toc

Example: Write a script called “time_keepign.m” to measure the time it takes to calculate factorial of numbers from 1 to n. Program should repeat this process 1000 times changing n from 1 to 1000. Then plot the time for each n end point.



⁶ <https://www.mathworks.com/help/matlab/ref/tic.html>

⁷ <https://www.mathworks.com/help/matlab/ref/toc.html>

BME1901 – Introductory Computer Sciences
Laboratory Handout – 7

PROBLEMS

1. Write a function called “bubble_sort” that accepts a vector input, then using bubble sort algorithm sorts the input in ascending order, and returns the sorted vector. (You may not use any built-in MATLAB functions.)

2. Write a script (m-file) called “sort_compare.m” that compares the sorting times of bubble_sort() function from question 1 and built-in MATLAB sort() function. Program should create a random vector and sort it in ascending order using both functions while keeping execution times for both. Then the program should print the vector length, time for bubble sort function and time for built-in sort function on the screen. The program should repeat this process for random vectors with length 10, 100, 1000 and 10000.