JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

EN.540.635
Software Carpentry

Lecture 11
Efficiency | Big O | Sorting Algorithm

Efficiency

- It is possible to write a program which could technically run forever, even though it is not technically wrong

- An efficient program is related to how the implemented algorithm uses computational resources:
    - Memory
    - Time

- You will not be writing your own algorithm; you will be implementing algorithms already developed for other problems

Order of Growth

- To compare the speed or time efficiency of algorithms, we might try comparing the time they take to complete

- The above metric is influenced by:
  - Speed of the machine
  - Python implementation
  - Input size

- Efficiency in the end depends on the number of steps needed to execute on a given size of input

Order of Growth

- How does a problem scale?

- Asymptotic growth models are used to calculate the growth, as it approaches a limit on the size of the input

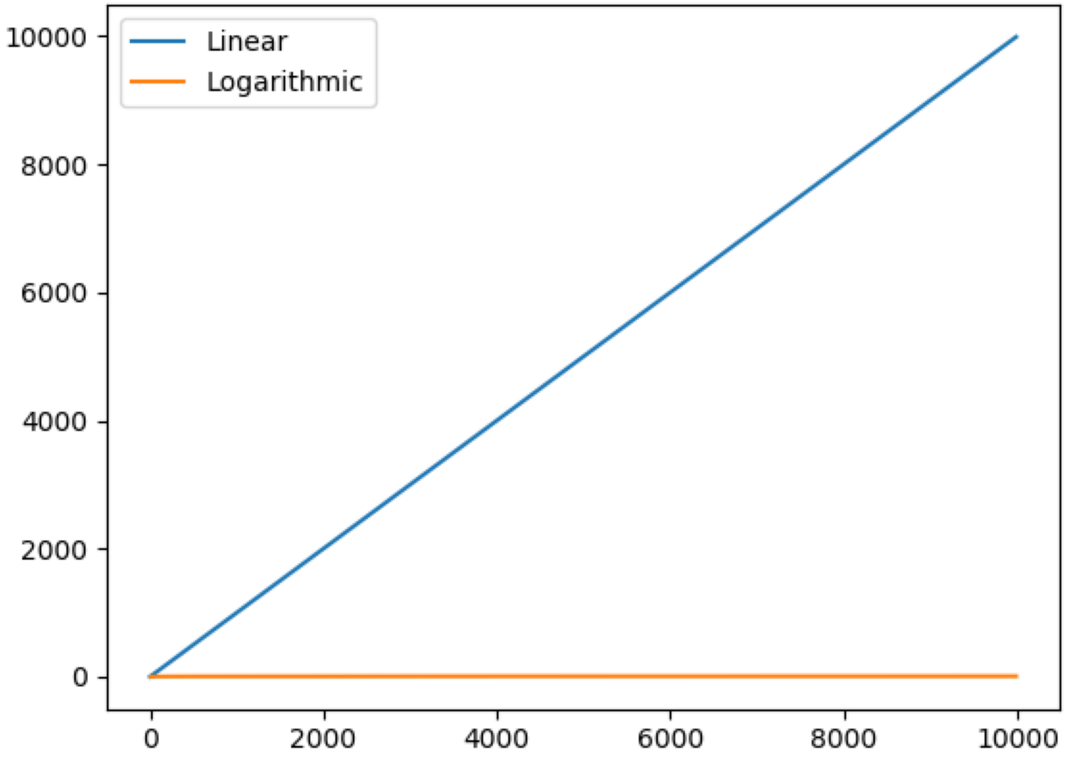
- Useful to define some parameter describing how something scales
  - O notation – Upper bound
  - Ω notation – Lower bound
  - Θ notation – Exactly bound

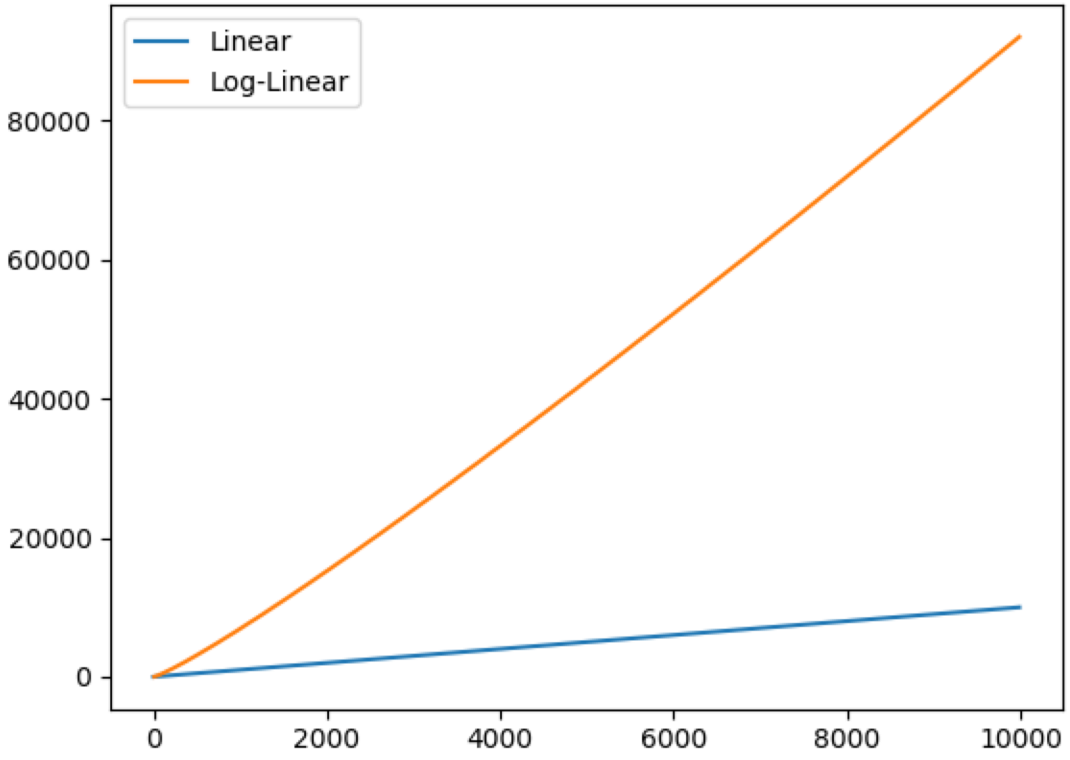
Big O Notation

- Worst case scenario and is the conventional notation used by computer scientists studying algorithms

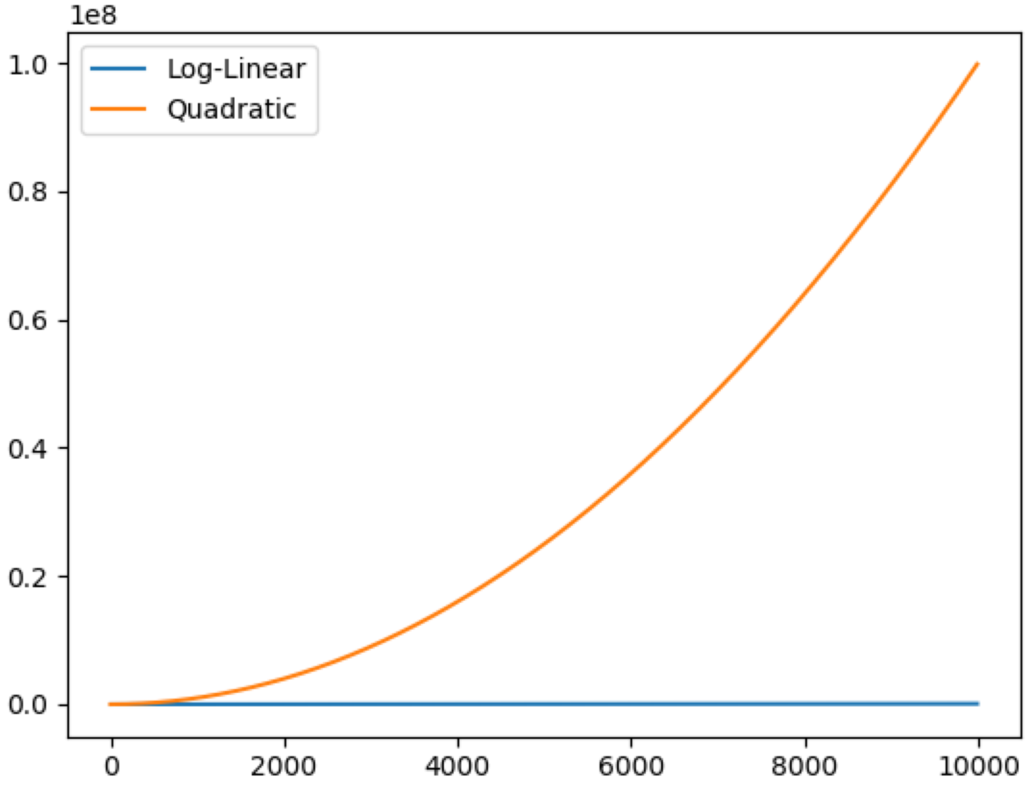
Fit Functions	Big O Notation
$y = 4 * x$	$O(N)$
$y = 12^{99999} * x$	$O(N)$
$y = 4 * \log_{10} 4x$	$O(\log(N))$
$y = x^4 + 12$	$O(N^4)$
$y = x * \log_{10}(x + 4) + x$	$O(Nlog(N))$

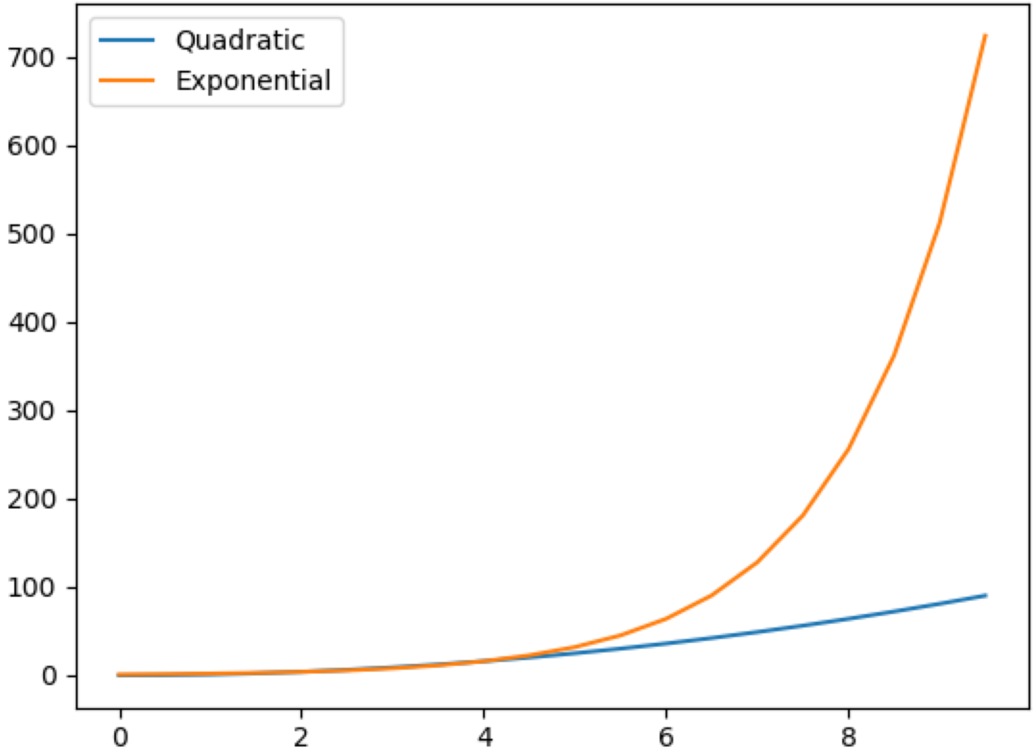
Examples


Linear
Logarithmic
10000
8000
6000
4000
2000
0
0
2000
4000
6000
8000
10000


Linear
Log-Linear
80000
60000
40000
20000
0
0
2000
4000
6000
8000
10000

Examples


1e8
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
Log-Linear
Quadratic


700
600
500
400
300
200
100
0
0
2
4
6
8
Quadratic
Exponential

Sorting Algorithms

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

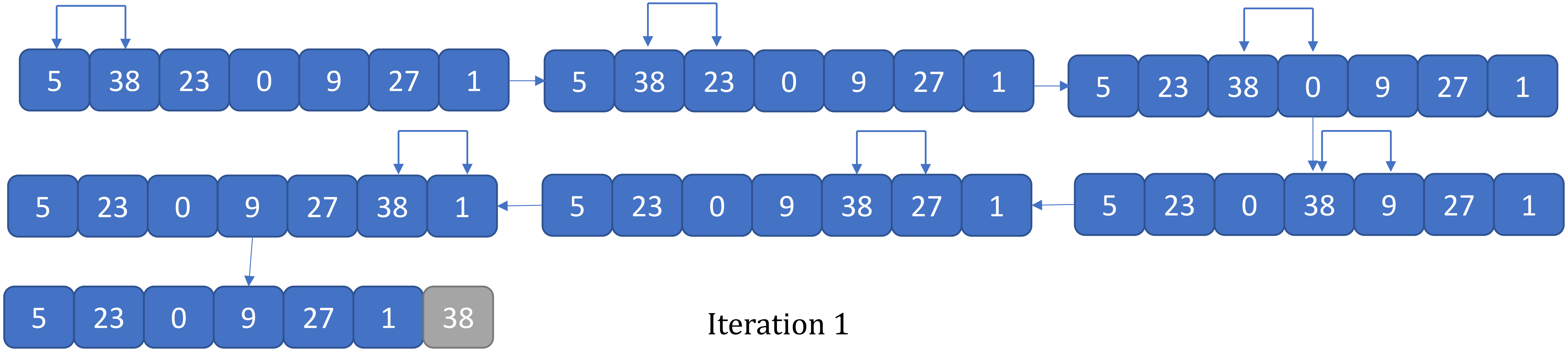
How can we sort the following array:

5	38	23	0	9	27	1

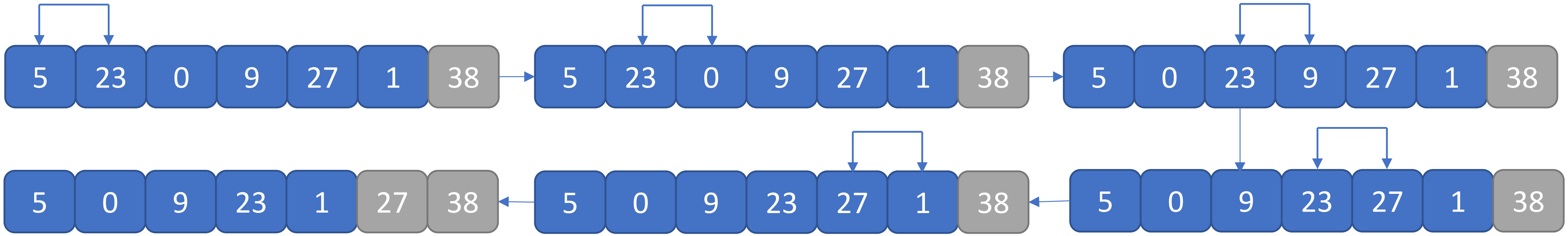
1. Insertion $O(N^2)$
2. Merge $O(Nlog(N))$
3. Quick $O(N^2)$
4. Bubble $O(N^2)$
5. Radix $O(wN) \rightarrow \mathrm{O(N), O(Nlog(N))}$, worse
6. More*

*Seizure Warning!

Bubble Sort

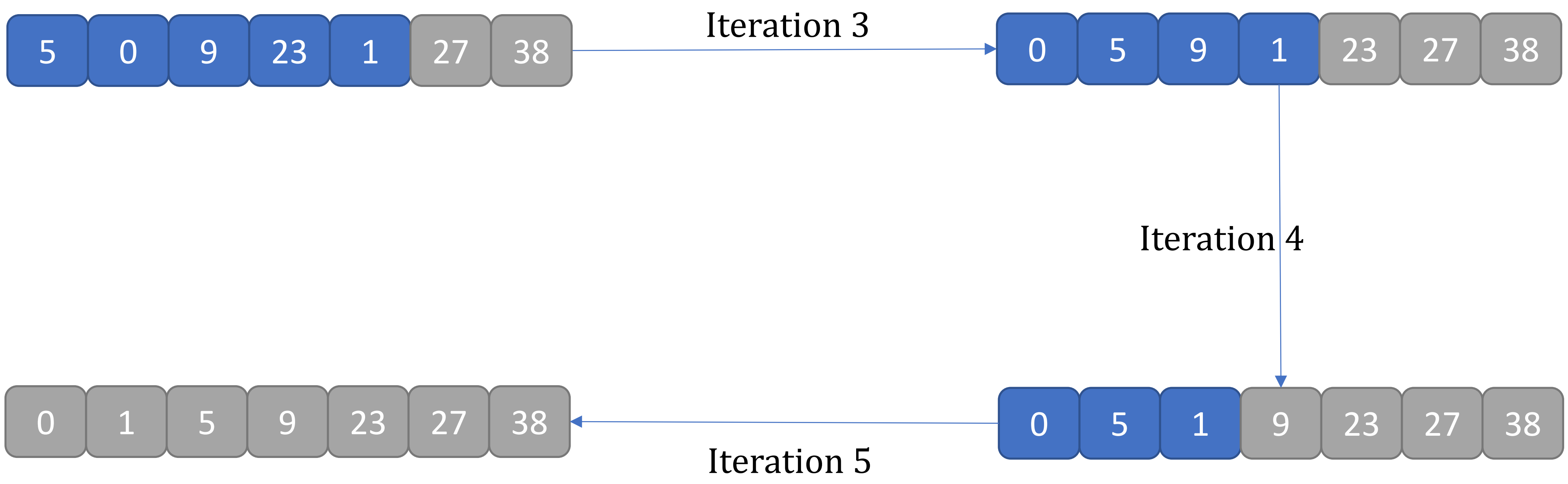


Iteration 1

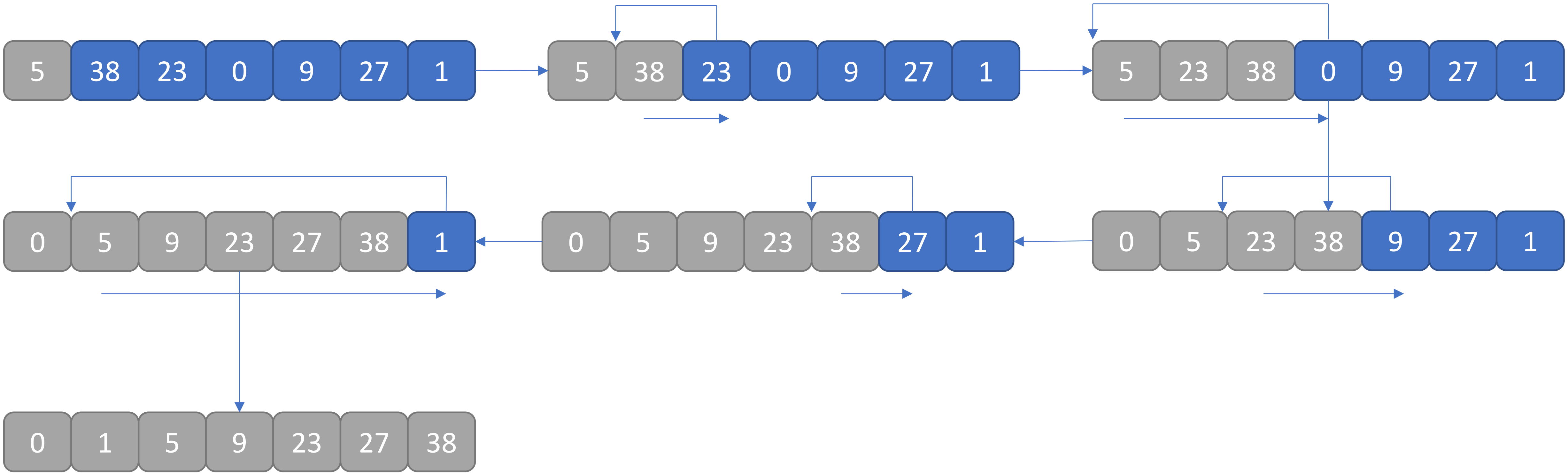


Iteration 2

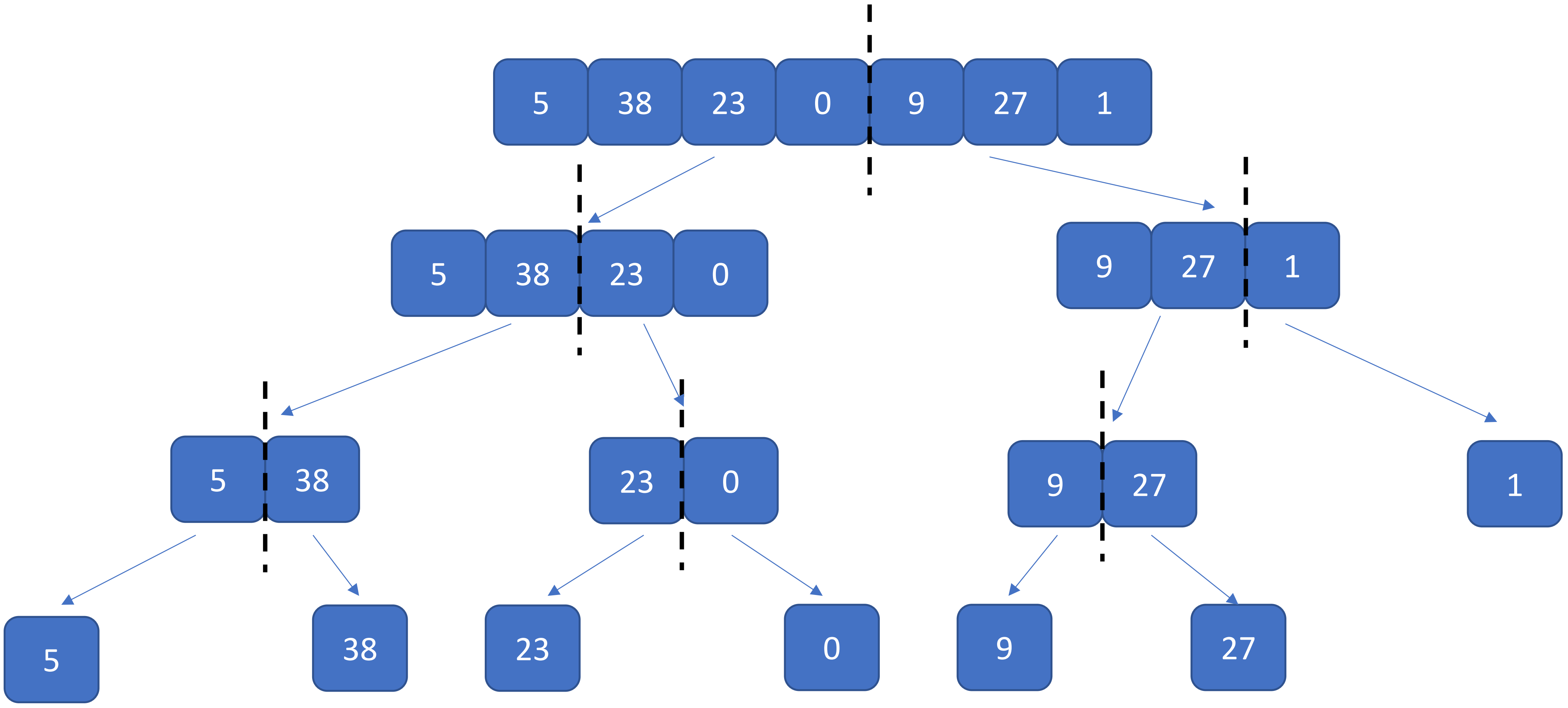
Bubble Sort


5 0 9 23 1 27 38
Iteration 3
0 5 9 1 23 27 38
Iteration 4
0 5 1 9 23 27 38
Iteration 5
0 1 5 9 23 27 38

Insertion Sort



Merge Sort



Merge Sort