Experimental Analysis of Searching Algorithms

01 Importance

Why searching algorithms are quite important?

02 Algorithms

Which searching algorithms have used for this experiment?

03 Methods

What methods have used to evaluate results?

04 Results

Which algorithm should be used in what cases?

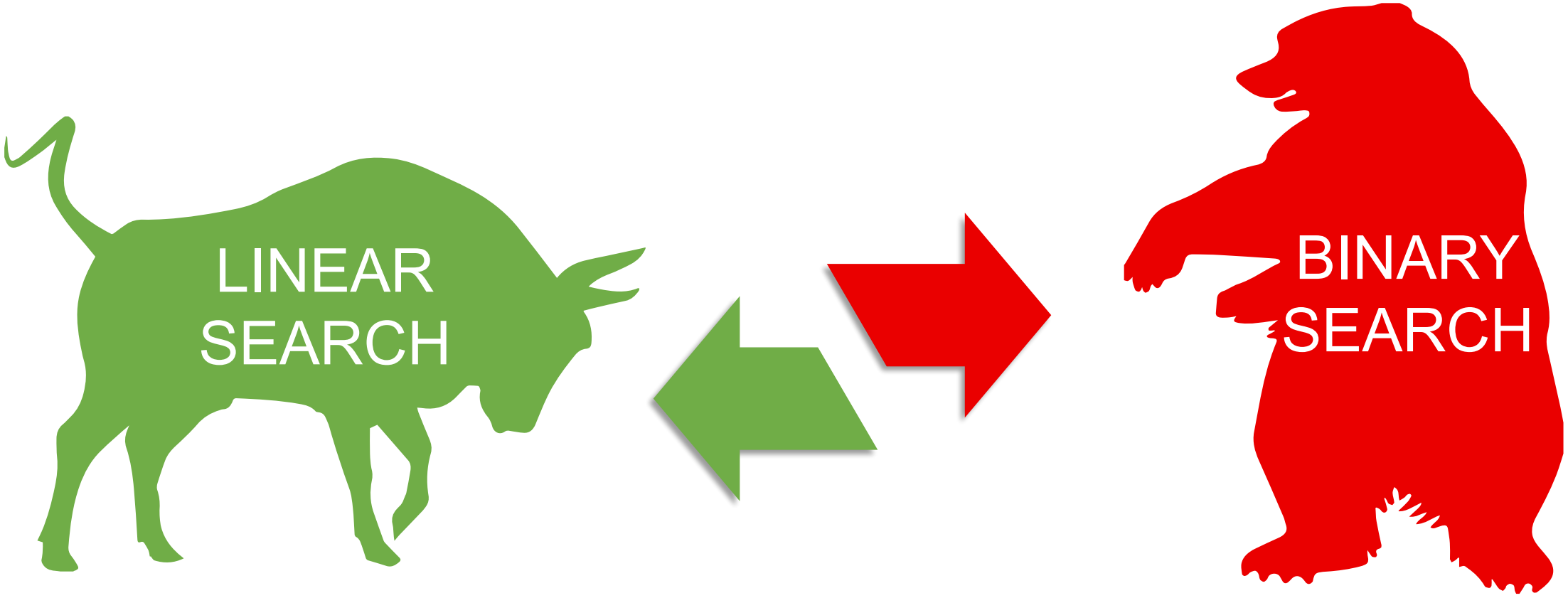
Importance of Searching Algorithms

Huge Data

Memory Usage

Response Time

Performance


LINEAR
SEARCH
BINARY
SEARCH

Linear Search

How it works?

In Linear search, we search an element or value in a given array by traversing the array from the starting, till the desired element or value is found.

Complexity

O(n)

When to use?

If the project requires to search this for very few times

Binary Search

How it works?

Searches a sorted array by repeatedly dividing the search interval in half. If the value is less than the item, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeat these steps.

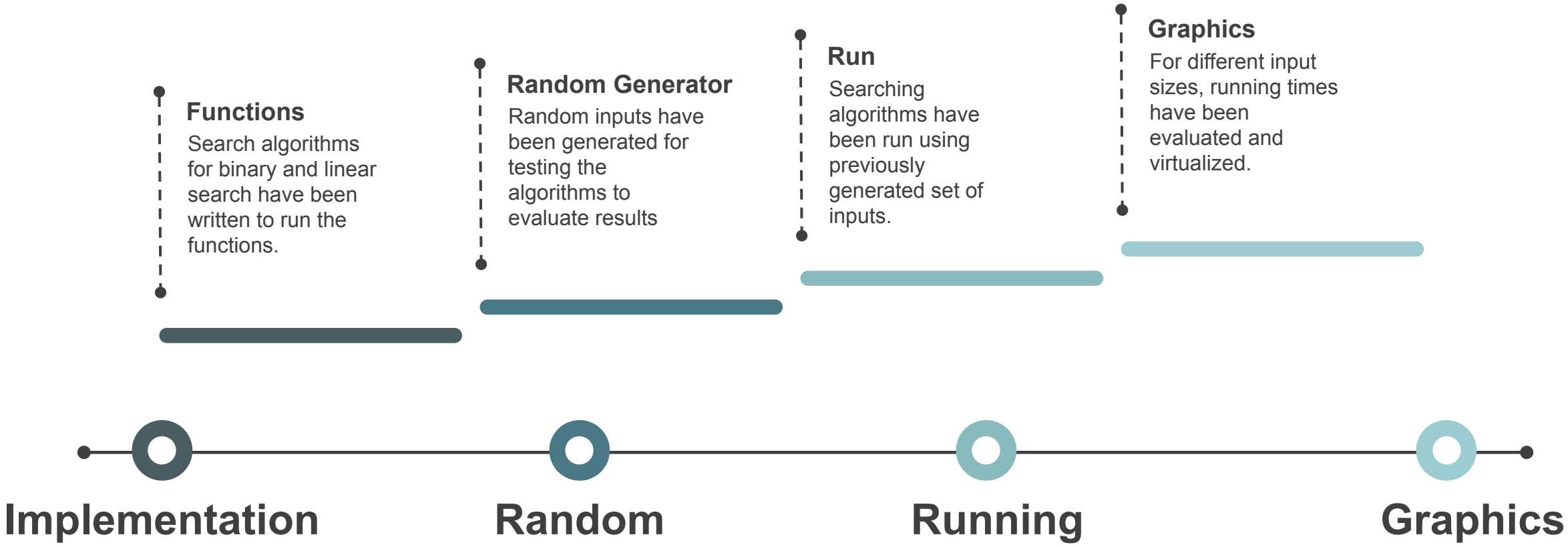
Complexity

O(n*log(n) + log(n)) = O(n*log(n)) (sorting the array)

When to use?

If the project requires to search this data for plenty of times

Methods


Functions
Search algorithms for binary and linear search have been written to run the functions.
Random Generator
Random inputs have been generated for testing the algorithms to evaluate results
Run
Searching algorithms have been run using previously generated set of inputs.
Graphics
For different input sizes, running times have been evaluated and virtualized.
Implementation
Random
Running
Graphics

RESULTS

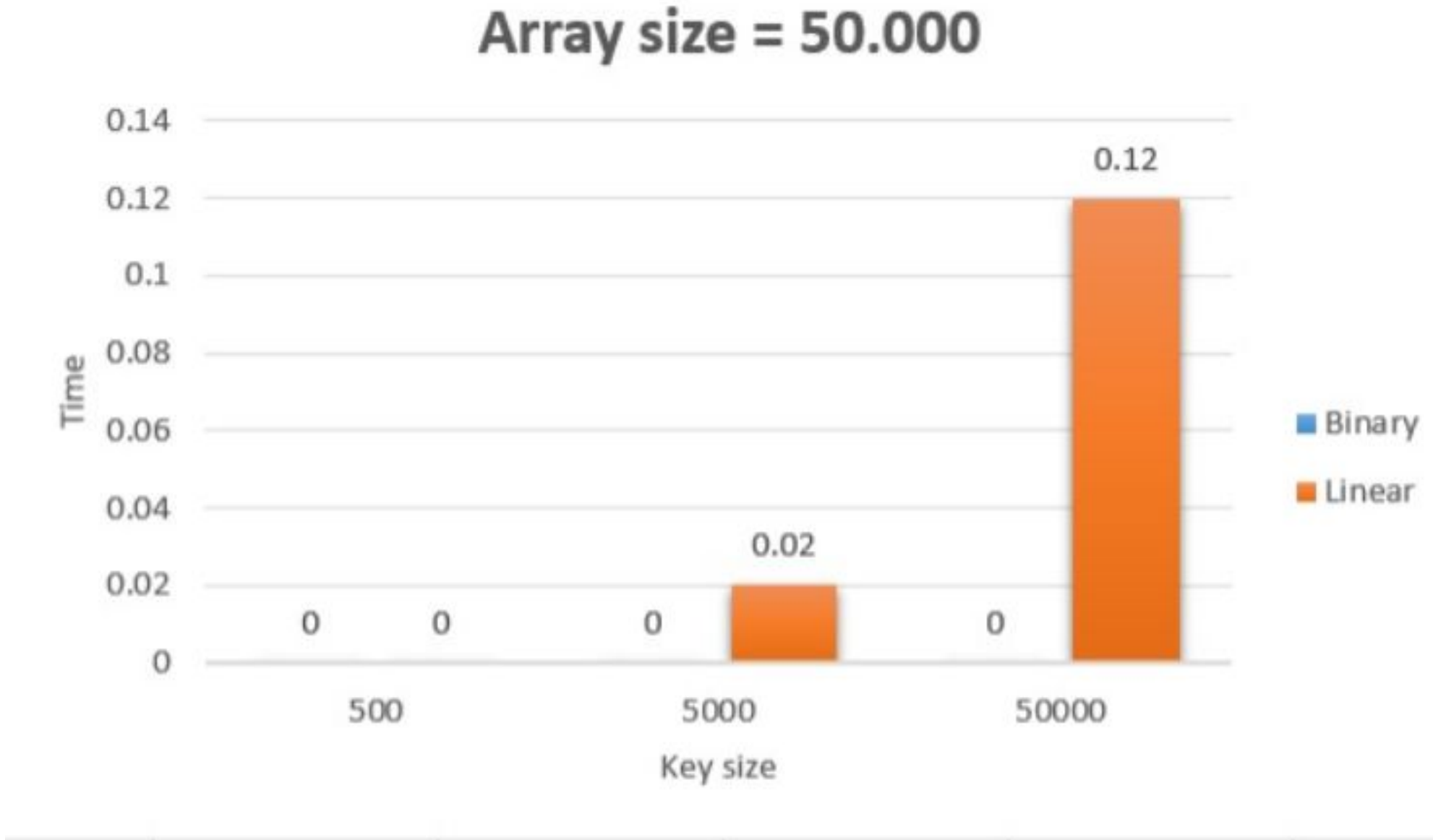
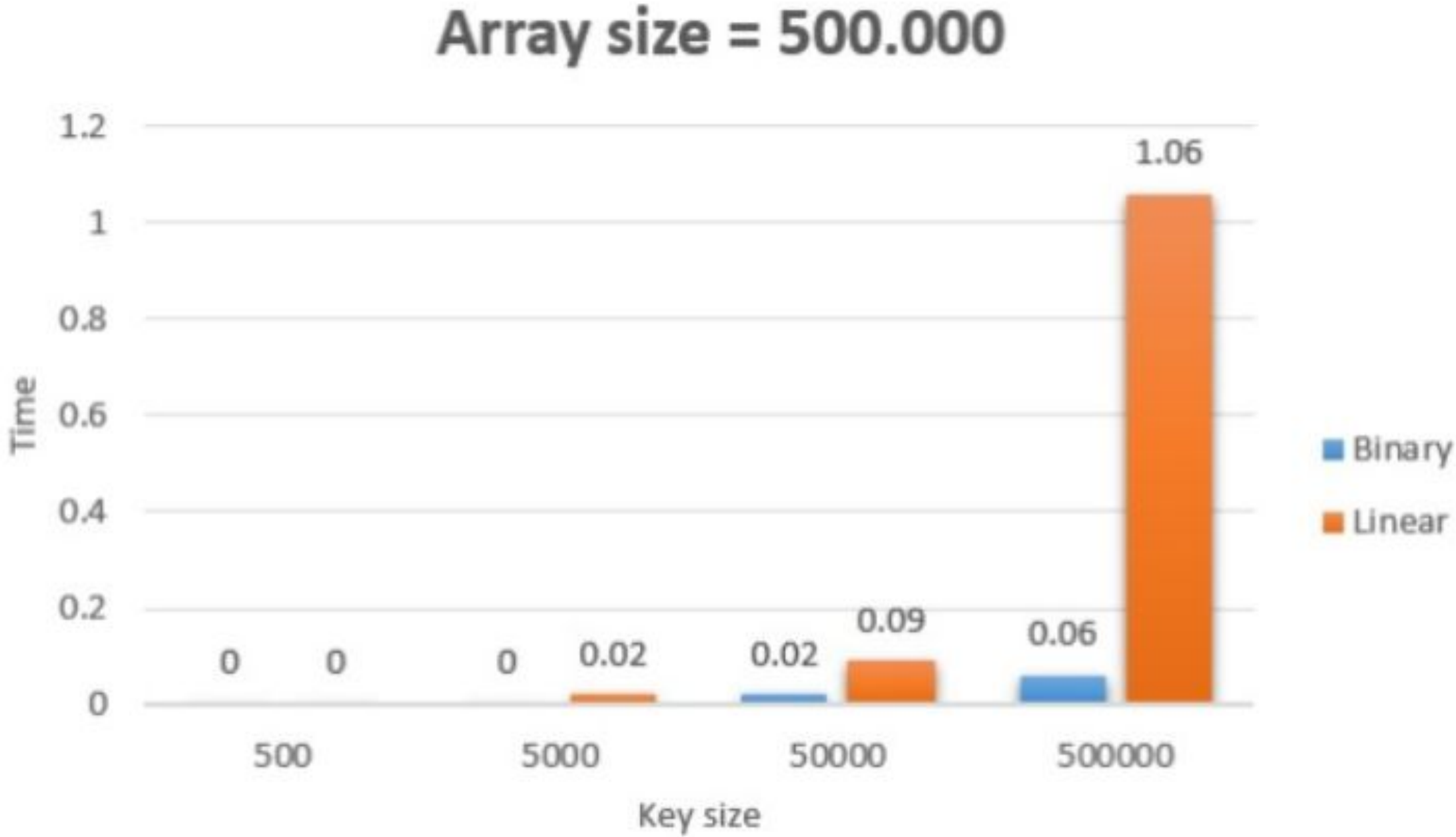
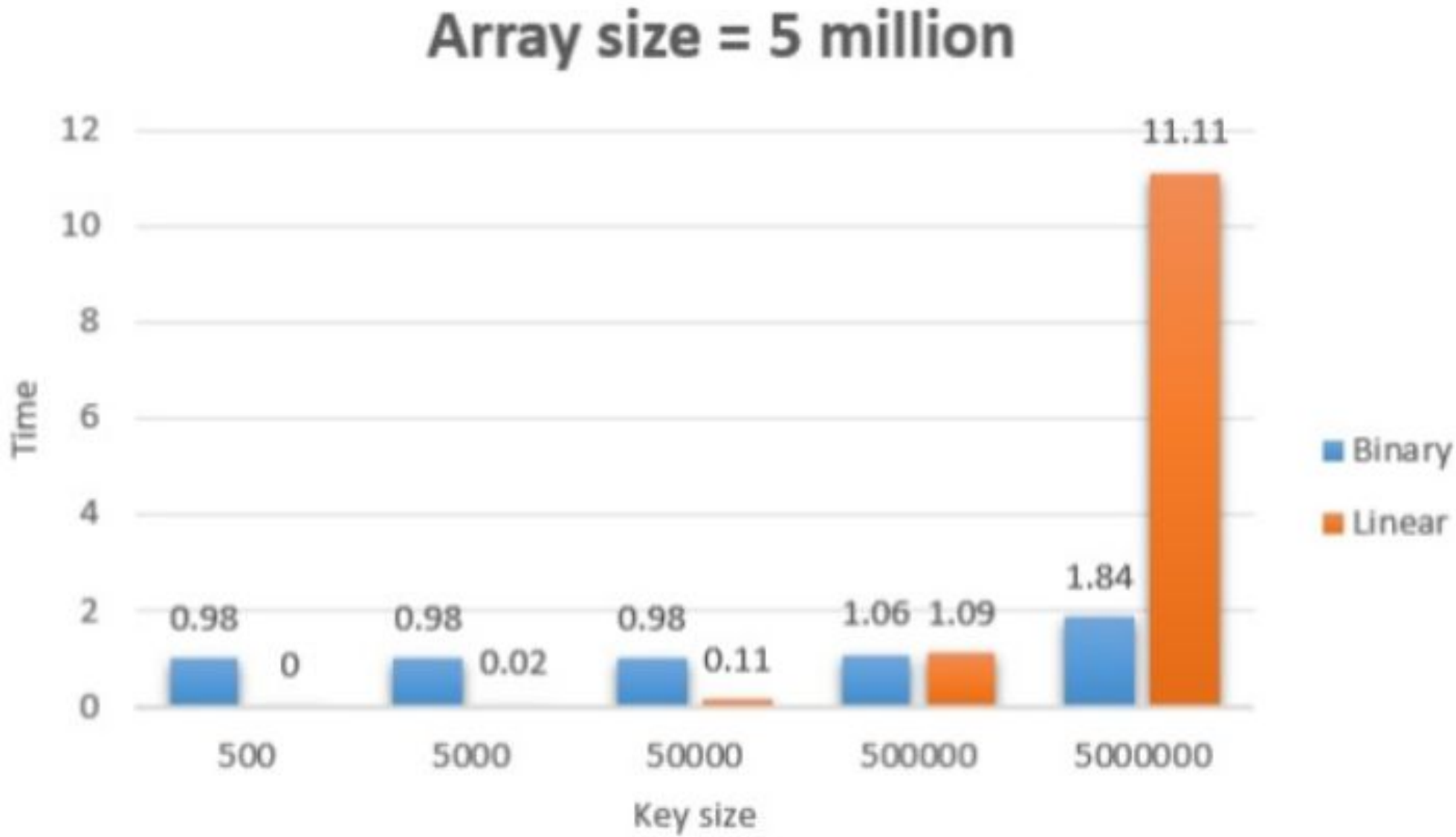


Chart for the running time for array size 50.000


Array size = 500.000
1.2
1
0.8
0.6
0.4
0.2
0
Time
1.06
0.09
0.06
0.02
0.02
0
0
0
500
5000
50000
500000
Key size
Binary
Linear



Time complexity for array size 5 million