УДК 811.111:004.05

BINARY SEARCH ALGORITHM

Oskirko A. V., student
Pogonya S. S., student
Scientific supervisor – Vanik I. Y., senior lecturer
English language department №1
Belarusian National University of Technology
Minsk, Republic of Belarus

The binary search algorithm is one of the most well-known algorithms for searching sorted data arrays. This is a method with logarithmic complexity that makes it possible to find an element in a sorted array in the minimum number of steps. Unlike linear search, which requires a complete search of the array elements, binary search works more efficiently, especially on large datasets. However, the sorting algorithm increases the overall time complexity.

Next, the example of pseudocode realization is shown:

```
Function BinarySearch(array, target):
   left = 0
right = size (of the array) - 1
   While left is less than right:
      middle = left + (right - left) / 2
If the array is[middle] == target:
Returns the average
      Otherwise, if the array is [average] < target:
left = average + 1
      More:
         right = middle - 1
   Return -1
```

The binary search algorithm works as follows: first you need to determine two pointers – to the beginning and the end of the array. After that, the index of the middle of the array is calculated. If the element in the middle is equal to the desired value, the search is completed and the index of this element is returned. If the element in the middle is smaller than the one you are looking for, the search continues on the right side of the array, and if it is larger, on the left. This process is repeated until ei-

ther the element is found or it turns out that it is missing from the array [1].

It is worth noting that the pre-sorting of the array is a prerequisite for using this algorithm. Because the search is only applicable in the case of ordered values. Binary search exists in two interpretations: iterative and recursive. The iterative model uses a loop in which the middle element is checked at each step, and then the search area is reduced. The recursive interpretation is that the algorithm calls itself for further search in one of the parts of the array. The time complexity of both implementations is O(log n), where n is the number of elements in the array. The number of search operations increases more slowly compared to linear search, the complexity of which is O(n) [2].

Moreover, the high speed of data processing is emphasized as a big advantage of this algorithm. When working with large arrays, it handles the task much faster than linear search. Binary search is also well suited for searching in datasets located in external memory, such as on a hard disk or in the cloud. The algorithm finds application in various fields of information technology. For example, it can be used to search a database. In industries such as computer graphics, it is used to search through color palettes. This is necessary for a simplified and ergonomic selection of colors. In machine learning, binary search is used in model parameter selection methods. Mastering the binary search algorithm is also essential for developers to improve their problem-solving skills [2].

In conclusion, it should be noted that the binary search algorithm is an important component in the field of technology. It provides faster access to any parameters, searches for information or values in large amounts of data, and helps simplify many information and computing processes. Despite the need to sort before using the algorithm, the advantages of binary search make it one of the most important tools in the development of automated programs.

References

1. Binary Search – Algorithm and Time Complexity Explained. – URL: https://www.freecodecamp.org/news/binary-search-algorithm-and-time-complexity-explained/ (date of access: 10.03.2025).

2. Binary Search Algorithm – Iterative and Recursive Implementation. – URL: https://www.geeksforgeeks.org/binary-search/ (date of access: 12.03.2025).