American Journal of
Electronics & Communication

# An API in JAVA Which Render Ease at Programming for Developers

Authors : Bhuvan Agarwal[*], Soumyajeet Bhattacharjee[*], Sima Kar[*], Madhurima Saha[*], Vijay Kumar[*], Dr. Sandip Mandal[*], UEM Kolkata
{Email: *sandy06.gcect@gmail.com*}

***Abstract*** - *Based on the concept of Application programming interface (API).This project comprises of a package named "algokit" which contains several algorithms based on the category of searching, sorting, dynamic programming, tree traversals and swapping. Keeping in mind that different algorithms from the same category have its own benefit in time and space complexity, This project covers almost all the algorithms known and available from each category. This would give the user several options to choose the right algorithm for its code.An user just requires to import the package named AlgoKit and call the functions inside it for a smooth programming experience. One of the prime objectives of this project is to build a kit that serves the purpose of reducing the number of lines of code and also reduce the time taken to run the same code elsewhere. It is platform independent and can be used in any open source Java development environment.*

***Keywords***: *API, AlgoKi, JAVA*

## INTRODUCTION

The modern era is the age of expertise and focuses on new development along with excelling in existing technologies. Today the world surrounding us is full of complex application and software. Development of these applications requires different kinds of algorithm in one project itself.

It is beneficial for developers to have an package which contains several important algorithms which is required in developing software and applications. For example-A software like Uber, which provides cab services has a very basic requirement of tracking locations on the map.

Apart from the pool of features that the software provides there must be a code for MST in that software. Keeping in mind the difficulties of a developer, Algokit is a package of algorithms which not only saves the CPU time for compiling but also saves the time and effort of a developer of explicitly writing a basic code.

## LITERATURE REVIEW

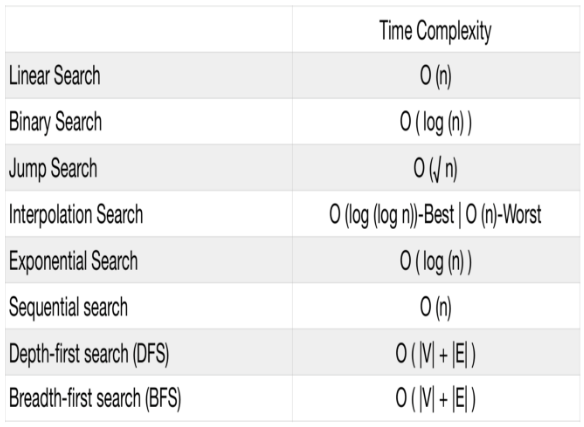
Java Application Programming Interface is a library of prewritten classes.The library is divided into packages and classes. It means one can either import a single class (along with its methods and attributes), or a whole package that contains all the classes that belong to the specified package.To use a class or a package from the library,one needs to use import keyword. Similarly, this ALGOKIT is a java API that is built with all the packages and classes of sorting, searching, graphs, matrices and exception classes.

By importing this package from ALGOKIT one can get all the necessary methods and classes required.

## PROPOSED WORK

<u>Searching Algorithms</u>: Every searching algorithm has its own best case. Focusing on that fact, here are the list of the algorithms which is present in the package.

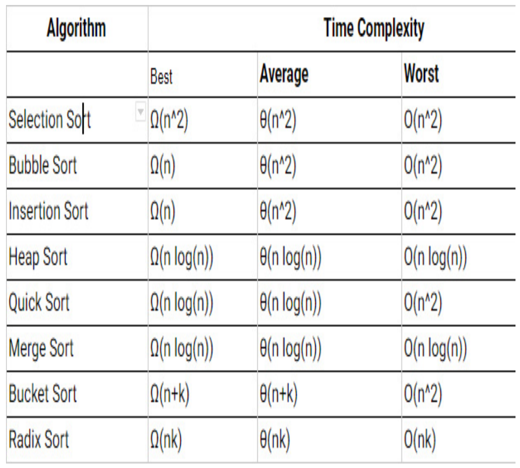
1. Binary Search
2. Exponential Search
3. Fibonacci Search
4. Interpolation Search
5. Jump Search
6. Linear Search
7. Ternary Search



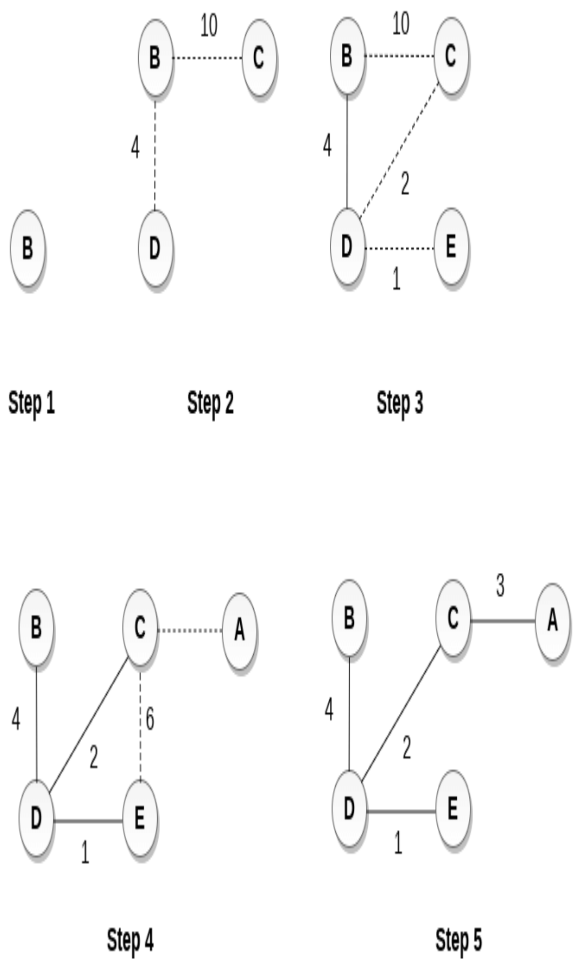
| | Time Complexity |
|---|---|
| Linear Search | O (n) |
| Binary Search | O ( log (n) ) |
| Jump Search | O (√ n) |
| Interpolation Search | O (log (log n))-Best \| O (n)-Worst |
| Exponential Search | O ( log (n) ) |
| Sequential search | O (n) |
| Depth-first search (DFS) | O ( \|V\| + \|E\| ) |
| Breadth-first search (BFS) | O ( \|V\| + \|E\| ) |

<u>Sorting algorithms</u>: A comparative study of the sorting techniques based upon time complexity is discussed. Some of the sorting techniques are as follows:

1. Bubble Sort
2. Counting sort
3. Insertion sort
4. Merge sort
5. Quick sort
6. Radix sort
7. Selection sort
8. Topological sort



| Algorithm | Time Complexity | | |
|---|---|---|---|
| | Best | Average | Worst |
| Selection Sort | Ω(n^2) | θ(n^2) | O(n^2) |
| Bubble Sort | Ω(n) | θ(n^2) | O(n^2) |
| Insertion Sort | Ω(n) | θ(n^2) | O(n^2) |
| Heap Sort | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) |
| Quick Sort | Ω(n log(n)) | θ(n log(n)) | O(n^2) |
| Merge Sort | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) |
| Bucket Sort | Ω(n+k) | θ(n+k) | O(n^2) |
| Radix Sort | Ω(nk) | θ(nk) | O(nk) |

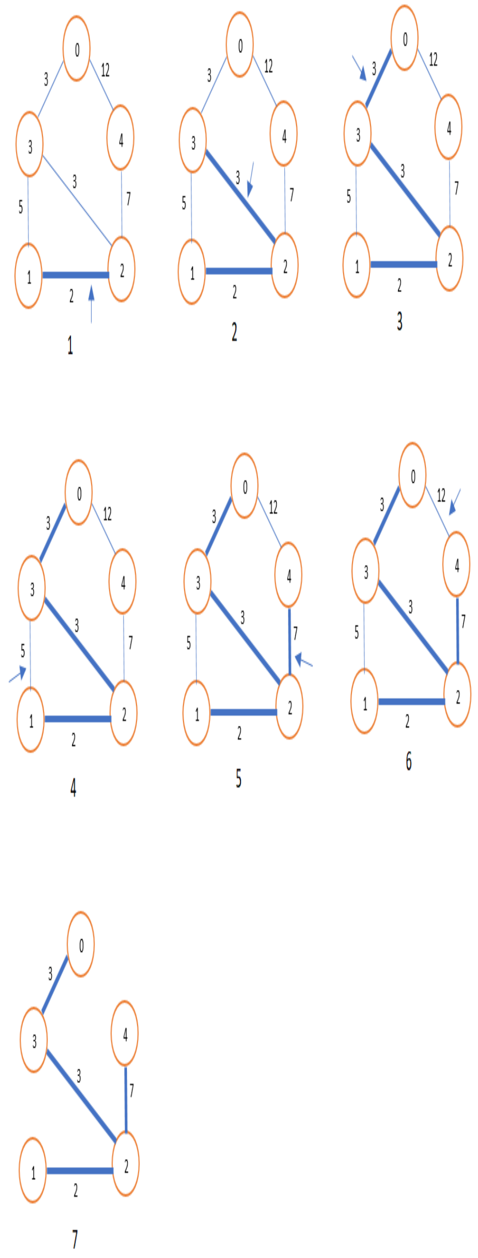


Matrix Manipulation: This topic forms a very important base for bigger graph problems and this class simple returns you the result of your matrices problems. From addition to finding Eigen values, some of the problems included are:

1. Matrix addition
2. Matrix subtraction
3. Matrix values

Graphs: MST Algorithms like Prims and krushkal returns the minimum spanning trees for a weighted, connected and undirected graph. To achieve the desired result, user only needs to give the graph as input and get the result by calling the function name like (graphs.Prims) and (graphs.Krushkal) commands respectively. Some of the graph related algorithms that this project includes are:

1. Prims
2. Krushkal
3. Dijsktras
4. DFS
5. BFS



## RESULT & ANALYSIS

We have tested the algorithms implemented in ALGOKIT API with sample data and compared It with the same algorithm explicitly written in the code and found that by using ALGOKIT API the task is fulfilled in lesser lines of code thereby saving time for developers and also it makes the code more readable.

Moreover, the time taken by the code to run is also comparatively lesser than equal to when we write the whole algorithm explicitly in the code.

As every algorithm implemented under ALGOKIT API falls under one category example – algokit.search , algokit.sort , algokit.graph etc. So maintenance of such an API is also very convenient and any future changes can be made effectively.

## CONCLUSION

This API contains implementation of various popular algorithms like searching , sorting , graph and matrix.It also contains several exception classes that can handle faulty data and wrong inputs. It comes in handy for developers to develop code faster just by calling these algorithms classes and its respective methods.

Thereby resulting in fewer lines of code and faster execution.

## FUTURE WORK

This project will further include more algorithms on Trees and graphs that comes in handy for real world use cases. Algorithms on Trie data structure will also be included in future.

## REFERENCES

[1] N. Srinivasan and A. Selvaraj, "Mobile based data retrieval using RDF and NLP in an efficient approach," *2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM),* Chennai, India, 2017, pp. 427-428, doi:10.1109/ICONSTEM.2017.8261416.

[2] Knuth, Donald (1997). "Section 6.1: Sequential Searching,". Sorting and Searching. The Art of Computer Programming. 3 (3rd ed.). Addison-Wesley. pp. 396–408. ISBN 0-201-89685-0.

[3] Thomas H. Cormen. Charles E. Leiserson. Ronald L. Rivest. Clifford Stein. Introduction to Algorithms. Third Edition. The MIT Press. Cambridge, Massachusetts.

[4] R. Boorugu and G. Ramesh, "A Survey on NLP based Text Summarization for Summarizing Product Reviews," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 352-356,doi:10.1109/ICIRCA48905.2020.9183355.

[5] K. Nokkaew and R. Kongkachandra, "Keyphrase Extraction as Topic Identification Using Term Frequency and Synonymous Term Grouping," *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Pattaya, Thailand, 2018, pp. 1-6, doi: 10.1109/iSAI-NLP.2018.8693001.

[6] N. Chumuang and M. Ketcham, "Model for Handwritten Recognition Based on Artificial Intelligence," *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Pattaya, Thailand, 2018, pp. 1-5, doi: 10.1109/iSAI-NLP.2018.8692958.

[7] M. R. Hasan, M. Maliha and M. Arifuzzaman, "Sentiment Analysis with NLP on Twitter Data," *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2),* Rajshahi, Bangladesh, 2019, pp. 1-4, doi: 10.1109/IC4ME247184.2019.9036670.

[8] M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques," *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, Anaheim, CA, USA, 2015, pp. 169-170,doi:10.1109/ICOSC.2015.7050801.

*University of Engineering and Management Kolkata