International Journal of Mathematical, Engineering, Biological and Applied Computing, 2022, 1, 8-11
www.scipublications.org/journal/index.php/ijmebac
DOI: 10.31586/ijmebac.2022.285

Article

A Linear-Time Algorithm to Find the Second Smallest Number

Raja Marappan *, S. Bhaskaran

School of Computing, SASTRA Deemed University, Thanjavur, India

*Correspondence: Raja Marappan (professor.m.raja@gmail.com)

Abstract: An algorithm is defined as a finite step-by-step procedure to accomplish a required result. It is also defined as a sequence of computational operations that convert the given input into the required output. In general, in the worst case, an algorithm is said to be optimal if there are no algorithms that perform a less basic number of well-defined operations, in the worst case. This paper presents an optimal algorithm for finding the second smallest among n numbers. The complexity of the proposed algorithm and its advantages are also analyzed in this paper.

Keywords: Algorithm, Computational Complexity, Optimal Algorithm, Second Smallest, Searching Smallest

How to cite this paper: Marappan, R., & Bhaskaran, S. (2022). A Linear-Time Algorithm to Find the Second Smallest Number. *International Journal of Mathematical, Engineering, Biological and Applied Computing*, *1*(1), 8–11. Retrieved from https://www.scipublications.com/journal/index.php/ijmebac/article/view/285

Received: March 22, 2022
Accepted: April 29, 2022
Published: May 01, 2022

1. Introduction

An algorithm is defined as an unambiguous procedure that contains a finite set of well-defined operations or steps that requires a finite amount of storage and takes a finite amount of time to complete and it can be executed by a mathematical machine [1]. The word 'Algorithmic' is defined as a branch of computer science that consists of both designing and analyzing algorithms. The word "design" pertains to the description of the algorithm at the abstract level through a pseudo-language. The algorithm's correctness is that it should solve the given problem in all cases. Next, the "analysis" part deals with the performance evaluation of the algorithm, also defined as the complexity analysis. This paper designs an optimal algorithm to find the second smallest among the given *n* numbers [2-4]. The complexity analysis and the advantages of the proposed algorithm are also explored in the next sections. Optimal algorithms should be designed to solve real-world problems either using hard computing or soft computing techniques [5-29].

2. Properties or characteristics of an Algorithm

The following are the important characteristics of every algorithm [3].

1. An algorithm must be precise.
2. An algorithm must be effective.
3. An algorithm must have a fixed finite number of instructions.
4. The execution of an algorithm must always terminate

3. Design of an optimal algorithm for finding the second smallest among *n* numbers

The optimal algorithm to find the second smallest among *n* numbers is designed to the properties mentioned in section 2 and the resultant optimal algorithm is presented in this section.

Algorithm Find-SecondMin (A[], *n*)
// A[1: *n*] is an one-dimensional array, that contains *n* elements
// Finding the second smallest among *n* elements with minimum comparisons, O(*n*)

```
fmin =A[1]; smin = A[2];
if (smin < fmin) then fmin = A[2]; smin = A[1];
for i = 3 to n do:
    if (fmin = smin) then smin=A[i];
    if (A[i] < fmin) then smin = fmin; fmin = A[i];
    else if ((fmin < A[i]) && (A[i] < smin)) then smin = A[i];
// End of Algorithm
```

4. Asymptotic Analysis of Algorithm Find-SecondMin (A[], *n*)

The best, worst, and average-case analysis of the devised algorithm is explained in this section [4,5].

Worst case: The total count of basic operations, that is, the count of comparisons performed in the worst case is given by

$w(n) = 1 + (n - 2 + 1)[2 + 2] = 1 + 4(n - 1) = 1 + 4n - 4 = 4n - 3 = O(n)$.

For example, n =5, and the inputs (80, 30, 35, 34, 32) result in worst-case performance.

Average case: The total count of basic operations, that is, the count of comparisons performed in the average case is given by

$A(n) = 1 + (n - 2 + 1)\ [1 + 1 + 1] = 1 + 3(n - 1) = 1 + 3n - 3 = 3\ n - 2 = O(n)$.

For example, n =5 and the inputs (10, 20, 30, 40, and 50) result in average-case performance.

Best case: The total number of basic operations, that is, the number of comparisons performed in the best case is given by

$f(n) = 1 + (n - 2 + 1)\ [1 + 1] = 1 + 2(n - 1\ ) = 1 + 2\ n - 2 = 2\ n - 1 = O(n)$.

For example, n = 5, and the inputs (10, 8, 5, 3, and 2) result in best-case performance.

5. Advantages of Algorithm Find-SecondMin (A[], *n*)

The advantages of the proposed algorithm are presented below.

1. It takes linear time complexity, O(n). Sorting and finding the 2nd smallest requires O(n^2) time complexity, hence it is an optimal algorithm for n (>=4) values.
2. The first smallest element is also computed and stored in the variable *fmin*. Hence applicable in determining the initial solution using the approximation method of Vogel.
3. If the first smallest, say x, appears in more than one location, and if the next smallest, say y, is available in the input set, then *fmin* = x and *smin* = y. If all inputs are the same, then *fmin* = *smin*.

6. Conclusions

In general, in the worst case, an algorithm x is said to be optimal if no algorithm computes less number of basic operations. Here we meant all possible available algorithms, including those not yet designed. Hence, the word “optimal” doesn’t mean “the best known”, it means “the best possible”. This paper explained the design of an optimal algorithm for finding the second smallest among n numbers. The computational complexity of the devised algorithm is analyzed for all three cases and its advantages are also discussed.

References

[1] Ellis Horowitz; Sartaj Sahni; Sanguthevar Rajasekaran: Fundamentals of Computer Algorithms. Galgotia Publications, 2003.

[2] Sara Baase; Allen Van Gelder: Computer Algorithms – Introduction to Design and Analysis. Third Edition, Pearson Education, 2008.

[3] T.H. Cormen; C.E. Leiserson; R.L. Rivest; C. Stein: Introduction to Algorithms. Second Edition, PHI, 2003.

[4] Aho; Hopcroft; Ullman: The Design and Analysis of Computer Algorithms. Fourth Impression, Pearson Education, 2009.

[5] Bhaskaran, S., Marappan, R. Design and analysis of an efficient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications. Complex Intell. Syst. (2021). https://doi.org/10.1007/s40747-021-00509-4

[6] Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. Mathematics 2021, 9, 197. https://doi.org/10.3390/math9020197

[7] Marappan, R., Sethumadhavan, G. Solving Graph Coloring Problem Using Divide and Conquer-Based Turbulent Particle Swarm Optimization. Arab J Sci Eng (2021). https://doi.org/10.1007/s13369-021-06323-x

[8] Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Comparative Analysis of New Personalized Recommender Algorithms with Specific Features for Large Scale Datasets. Mathematics 2020, 8, 1106. https://doi.org/10.3390/math8071106

[9] Marappan, R.; Sethumadhavan, G. Complexity Analysis and Stochastic Convergence of Some Well-known Evolutionary Operators for Solving Graph Coloring Problem. Mathematics 2020, 8, 303. https://doi.org/10.3390/math8030303

[10] N. S. Anand, R. Marappan and G. Sethumadhavan, "Performance Analysis of SAR Image Speckle Filters and its Recent Challenges," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2018, pp. 1-4, doi: 10.1109/ICCIC.2018.8782425.

[11] Marappan, R., Sethumadhavan, G. Solution to Graph Coloring Using Genetic and Tabu Search Procedures. Arab J Sci Eng 43, 525–542 (2018). https://doi.org/10.1007/s13369-017-2686-9

[12] R. Marappan and G. Sethumadhavan, "Solving channel allocation problem using new genetic algorithm with clique partitioning method," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, pp. 1-4, doi: 10.1109/ICCIC.2016.7919671.

[13] R. Marappan and G. Sethumadhavan, "Solution to graph coloring problem using divide and conquer based genetic method," 2016 International Conference on Information Communication and Embedded Systems (ICICES), 2016, pp. 1-5, doi: 10.1109/ICICES.2016.7518911.

[14] R. Marappan and G. Sethumadhavan, "Divide and conquer based genetic method for solving channel allocation," 2016 International Conference on Information Communication and Embedded Systems (ICICES), 2016, pp. 1-5, doi: 10.1109/ICICES.2016.7518914.

[15] Raja Marappan, Gopalakrishnan Sethumadhavan: Solving Fixed Channel Allocation using Hybrid Evolutionary Method. MATEC Web of Conferences 57 02015 (2016) DOI: 10.1051/matecconf/20165702015

[16] Marappan, R., & Sethumadhavan, G. (2015). Solving graph coloring problem for large graphs. Global Journal of Pure and Applied Mathematics, 11(4), 2487-2494.

[17] Marappan, R., & Sethumadhavan, G. (2015). Solution to Graph Coloring Problem using Evolutionary Optimization through Symmetry-Breaking Approach. International Journal of Applied Engineering Research, 10(10), 26573-26580.

[18] Marappan, R., & Sethumadhavan, G. (2015). Solution to graph coloring problem using heuristics and recursive backtracking. International Journal of Applied Engineering Research, 10(10), 25939-25944.

[19] G. Sethumadhavan and R. Marappan, "A genetic algorithm for graph coloring using single parent conflict gene crossover and mutation with conflict gene removal procedure," 2013 IEEE International Conference on Computational Intelligence and Computing Research, 2013, pp. 1-6, doi: 10.1109/ICCIC.2013.6724190.

[20] R. Marappan and G. Sethumadhavan, "A New Genetic Algorithm for Graph Coloring," 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation, 2013, pp. 49-54, doi: 10.1109/CIMSim.2013.17.

[21] Raja Marappan, Gopalakrishnan Sethumadhavan, R.K. Srihari, New approximation algorithms for solving graph coloring problem – An experimental approach, Perspectives in Science, Volume 8, 2016, Pages 384-387, ISSN 2213-0209, https://doi.org/10.1016/j.pisc.2016.04.083.

[22] Raja Marappan, Gopalakrishnan Sethumadhavan, U. Harimoorthy, Solving channel allocation problem using new genetic operators – An experimental approach, Perspectives in Science, Volume 8, 2016, Pages 409-411, ISSN 2213-0209, https://doi.org/10.1016/j.pisc.2016.04.091.

[23] S. Balakrishnan, Tamilarasi Suresh, Raja Marappan: Analysis of Recent Trends in Solving NP Problems with New Research Directions Using Evolutionary Methods. International Journal of Research Publication and Reviews Vol (2) Issue (8) (2021) Page 1429-1435

[24] S. Bhaskaran; Raja Marappan: New Personalized Recommendation System for E-Learning. AshEse Journal of Physical Science. Vol. 5(5), pp. 063-067, August, 2021 ISSN: 2059-7827 DOI: http://www.ashese.co.uk/ajps-v5-issue-5/new-personalized-recommendation-system-for-e-learning

[25] S. Balakrishnan, Tamilarasi Suresh, Raja Marappan. (2021) A New Multi-Objective Evolutionary Approach to Graph Coloring and Channel Allocation Problems. Journal of Applied Mathematics and Computation, 5(4), 252-263. DOI: http://dx.doi.org/10.26855/jamc.2021.12.003

[26] Raja Marappan: A New Multi-Objective Optimization in Solving Graph Coloring and Wireless Networks Channels Allocation Problems. Int. J. Advanced Networking and Applications Volume: 13 Issue: 02 Pages: 4891-4895 (2021)

[27] Raja Marappan, S. Bhaskaran, N. Aakaash, S. Mathu Mitha. (2022) Analysis of COVID-19 Prediction Models: Design & Analysis of New Machine Learning Approach. Journal of Applied Mathematics and Computation, 6(1), 121-126. DOI: http://dx.doi.org/10.26855/jamc.2022.03.013

[28] Raja Marappan, S. Bhaskaran, S. Ashwadh, H. Aathi Raj. (2022) Extraction of Drug Review Polarity Using Sentimental Analysis. Journal of Applied Mathematics and Computation, 6(2), 167-177. DOI: http://dx.doi.org/10.26855/jamc.2022.06.001

[29] Marappan, R., & Bhaskaran, S. (2022). Analysis of Network Modeling for Real-world Recommender Systems. International Journal of Mathematical, Engineering, Biological and Applied Computing, 1(1), 1–7.