Divide and Conquer Algorithms Theory

Nur Aimielia binti Rahim^{#1}, Mohamed Faidz Mohamed Said^{#2}

Universiti Teknologi MARA 70300 Seremban Negeri Sembilan, MALAYSIA ¹ nuraimieliarahim@gmail.com ² faidzms@ieee.org

Abstract— Divide and Conquer is an algorithms design paradigm based on multi-branches recursion. This algorithm works by recursively breaking down a problem into two or more subproblems, until these become simple enough to be answered directly. The arrangement common to a class of Divide and Conquer algorithms is represented by a program scheme which a theorem is presented that relates the functionality of this algorithm based on its structure and the functionalities of its sub algorithms. Various strategies for planning these algorithms ascend from this theorem and they are used to formally derive algorithms for sorting a list of numbers, forming the Cartesian product of two sets, and finding the convex hull of a set of planar points. Lastly, these algorithms had many beneficial qualities that make it a very significant algorithms design paradigm.

Keywords: Divide, Conquer, algorithm, recursion, branches

I. INTRODUCTION

Divide and Conquer is an algorithm design based on multibranches recursion. A Divide and Conquer algorithm works by recursively breaking down a problem into two or more subproblems of the related or same type, until these become simple enough to be answered directly. The solutions to the sub-problems are combined and give a solution to the original problem. Comprehension and arranging separate and overcome calculations is a mind boggling aptitude that requires a decent comprehension of the nature that identify with issue to be unraveled. When proving a theorem by induction, it is often necessary to replace the original problem with complicated problem or a more general in order to initialize the recursion. There is no systematic method for finding the proper generalization by [4]. These Divide and Conquer problems are seen when optimizing the calculation of a Fibonacci number with efficient double recursion. In this examination, they were centering from article name's Divide and Conquer calculations. This report was s research related calculations which is Divide and Conquer strategies. Additionally, this exploration was to formalize the reasons for calculation plan. This examination was behavior not just to get a more profound and more grounded comprehension of the calculations. In [4] they introduce a class of system plans which agreeably give a typical structure to communicating Divide and Conquer calculations. This depends on a perspective of these calculations as homomorphism between polynomial math on their yield and info space. In the current paper they confine the thought to and import subclass of all gap and vanquish calculation. Additionally, this hypothesis likewise utilized as a part of parcel of extent of life, for example, in industry application. Paper [8] presents a numerically stable algorithm and fast for computing the eigendecomposition of a symmetric block diagonal plus semi-separable matrix. They report numerical experiments that specify that the algorithm is significantly more rapidly than the standard method which treats the given matrix as a general symmetric thick network. Separation and vanquish are a technique to partition the log into pieces, by method for projection. The strategy selects a gathering of occasion firmly related in the log when the deteriorations systems were connected. Alternative advantage of this method, which is useful in the context of the Divide and Conquer approach is proposed in this research, is the ability to use various strategies to subdivide the input data to speed up the learning process by [10] and [11]. The methods have been advanced for the computation of the eigenvalues of such a matrix from [12] that present a Divide-and-Conquer approach to the computation of the eigenvalues of a symmetric tri-diagonal matrix through the evaluation of the characteristic polynomial. However, by [13] each college class on calculations shows this strategy as one of its essential to take care of an issue rapidly. For instance, recursively understands each of these, precisely parts the issue into two sub-issues, each significantly littler than the first, and after that pieces together the answer for every part into the general arrangement. This technique has likewise been utilized as a part of the outline of heuristics for NP-hard streamlining issues, however as of late, the heuristics planned along these lines were either excessively troublesome, making it impossible to investigate, or had greatly poor execution ensured.

II. BACKGROUND

In this research, the objective is to simplify the process of designing algorithms for specification. A Divide and Conquer algorithms were created years ago. It was a technique on how to solve problem, for example in industry. This technique has many methods and advantages for each solution. In this research, the method used is looping coding. Besides that, matrix and fuzzy method can also be used to solve this problem.

III. LITERATURE REVIEW

This section shows some of the work that have been carried out by other researchers in the application of Divide and Conquer algorithms. The research by [5] was behavior to discover calculation speculations speak to the structure regular to a class of calculations, for example, partition and-vanquish or backtrack. A calculation hypothesis for a class A gives the source to outline procedures particular strategies for planning calculations from formal issue details. They clear up this technique with late work on the hypothesis of worldwide inquiry calculations and quickly say few others. A few configuration techniques have been executed in the KIDS/CYPRESS framework and have been utilized to semi naturally determine numerous calculations.

A plan for designing Divide-and-Conquer algorithms by [6] that was initially exhibited in a past article is augmented. It connected to some new issues. The augmentation includes techniques for conforming the first particular in light of particular sorts of disappointments which can happen amid the configuration procedure. They infer a few Divide and Conquer algorithms that are impressively more effective than already. This exploration additionally highlights the expectation with which isolate and vanquish calculations can be changed over into a parallel arrangement.

Approaches for automatic presentation of relational information in the form of directed graphs are discussed in this research. Furthermore, the study conducted by an impression of the diagram format issue is introduced alongside an outline [7] of a few distinctive design calculations. A gap and overcome format calculation called Compoze is depicted in point of interest. This calculation creates particularly great results on extensive charts of a few hundred vertices. In this research paper [8], Divide and Conquer algorithms are used for computing the eigendecomposition of A. Thus, matrices in this form appear when one discretizes the Green's function for the two points value boundary problem.

The goal of Process Mining by [9] is to separate the models from logs of a framework. The hypothesis of districts can be utilized to change over a log into a Petri net, yet unfortunately the change requires calculations with high unpredictability. Petri net is an immaculate route because of graphical representation, expressive power and clear semantics. This paper gives different strategies to beat this impediment. Either by bunching occasions in the log and taking a shot at projections or by utilizing decay systems can be utilized to grow the pertinence.

This paper by [13] Divide-and-Conquer is one of the simplest techniques in the design and analysis of algorithms, and yet, until recently, there have been no performance sureties known for approximation algorithms based on this approach. The paper chapter presents approximation algorithms for various NPhard graph-theoretic cut problems, and their subsequent application as a tool for the "Divide" part for a wide variety of problems.

IV. METHODOLOGY

Divide and Conquer is based on the idea where the entire problem that is to be solved may be too large to understand or solve at once. The numbers break up unto small parts and solve them separately and combine it together. In this strategy, a problem is solved by integrating their solution into a solution for the whole problem. The sub problem can in turn be solved using Divide and Conquer algorithms or solved directly. This selection sort algorithm is presented. The function Select fulfils the specification SELECT that discussed in previous section evaluates as follows. The list (30,10,35,60,88,22,63)

Select:

(30,10,35,60,88,22,63)

=Compose o (Id x Select) o FirstRest (30, 10,35,60,88,22,63)

= Compose o (Id × Select):(30,10,35,60, (,88,22,63))

= Compose "(30,10, (35,60, (88,22.(63))))

=(10, C o n s ' (30(, (35,60, (88,22.(63)))) = (10 (30,35,60, 88,22.63)

Repeat the similar step to find the max number. Sort when applied to (30, 10,35,60,88,22,63) evaluates as follows:

Ssort " (30, 10, 35, 60, 88, 22, 63)

= Cons o (id x Ssort) o Select (30, 10, 35, 60, 88,22,63)

= Conso (Id x Ssort) : (30,10, (35,60,

(88,22.(63)))

= Cons: (30,10 (35,0, (88,22(63))))

=(30,10,35,60,88,22,63)

where Ssort" (30(, (35,60, (88,22.(63))) evaluates to (30,35,60, 88,22.63) in a similar manner. Ssort : Xo =- if Xo = ni then nil

> else Cons o (Id x Ssort) o Select : xo then FirstRest : x else Compose o (Id x Select)o FirstRest : x Compose: (vl, (v2, z)) – if vl <~ v2

then (vl, Cons: (v2, z)) else (v2, Cons: (vl, z)) $\label{eq:constraint}$

V. CONCLUSION

In conclusion, the Divide and Conquer algorithms had many benefit qualities that make it a very significant algorithms design paradigm. It includes three simple steps to make easier, and those steps are to divide the complex problem into two or more sub problems, recursively solve the sub problem and conquer and combine the solution to the original problems. It is recommended for other to use this technique because of its simplicity and suitability.

REFERENCES

- Smith, D. R. (1985). The Design of Divide and Conquer Algorithms. Science of Computer Programming, 5, 3754-5158
- [2] Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983) Data Structures and Algorithms.
- [3] J. Bentley. (1980), Multidimensional Divide and Conquer, Comnt ACM 23 214-229.
- [4] D.R. Smith, (1983). The Structure of Divide and Conquer Algorithms, Technical Report NPS 52-83-002, Naval Postgraduate School, Monterey, CA
- [5] Smith, D. R., & Lowry, M. R. (1990). Algorithm Theories and Design Tactics. Science of computer programming, 14(2), 305-321.
- [6] Smith, D. R. (1987). Applications of a Strategy for Designing Divideand-Conquer Algorithms. Science of Computer Programming, 8(3), 213-229.

- [7] Messinger, E. B., Rowe, L. A., & Henry, R. R. (1991). A Divide-and-Conquer Algorithm for The Automatic Layout of Large Directed Graphs. Systems, Man and Cybernetics, IEEE Transactions on, 21(1), 1-2
- [8] Chandrasekaran, S., & Gu, M. (2004). A Divide-and-Conquer Algorithm for The Eigendecomposition of Symmetric Block-Diagonal plus Semiseparable Matrices. Numerische Mathematik, 96(4), 723-731.
- [9] Carmona, J., Cortadella, J., & Kishinevsky, M. (2009). Divide-and-Conquer Strategies for Process Mining. In Business Process Management (pp. 327-343). Springer Berlin Heidelberg.
- [10] Stach, W., Kurgan, L., Pedrycz, W., & Reformat, M. (2005). Genetic Learning of Fuzzy Cognitive Maps. Fuzzy sets and systems, 153(3), 371-401.
- [11] Stach, W., Kurgan, L., Pedrycz, W., & Reformat, M. (2004). Learning Fuzzy Cognitive Maps with Required Precision using Genetic Algorithm Approach.Electronics Letters, 40(24), 1.
- [12] Krishnakumar, A. S., & Morf, M. (1986). Eigenvalues of a Symmetric Tridiagonal Matrix: A Divide and Conquer Approach. Numerische Mathematik,48(3), 349-368.
- [13] Shmoys, D. B. (1997). Cut Problems and Their Application to Divideand-Conquer. Approximation algorithms for NP-hard problems, 192-235.