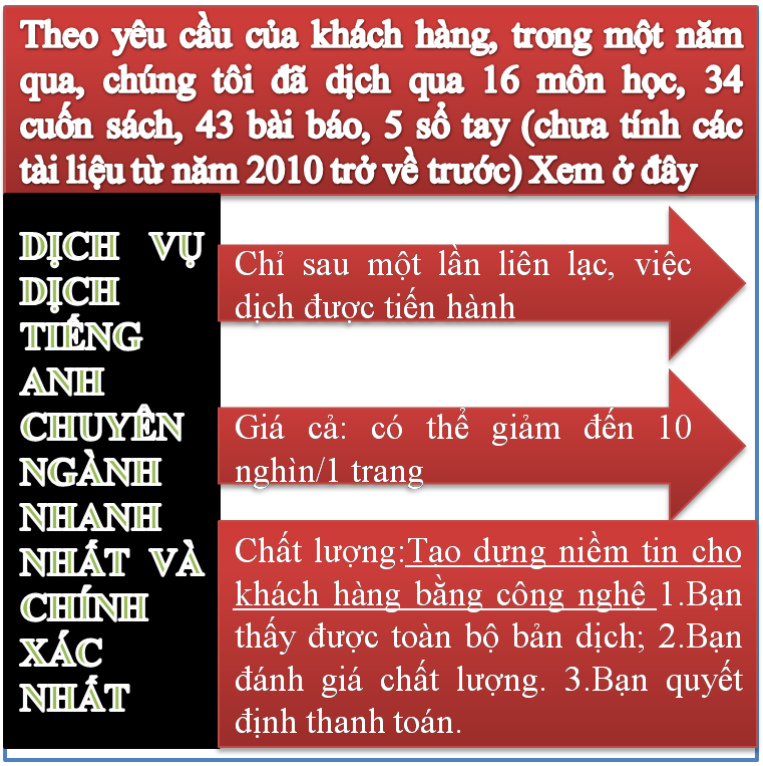



**Tài liệu này được dịch sang tiếng việt bởi:**

**www.mientayvn.com**

**Tìm bản gốc tại thư mục này (copy link và dán hoặc nhấn Ctrl+Click):**

**https://drive.google.com/folderview?id=0B4rAPqlxIMRDSFE2RXQ2N3FtdDA&usp=sharing**

**Liên hệ để mua:**

thanhlam1910_2006@yahoo.com hoặc frbwrthes@gmail.com hoặc số 0168 8557 403 (gặp Lâm)

**Giá tiền: 1 nghìn /trang đơn (trang không chia cột); 500 VND/trang song ngữ**

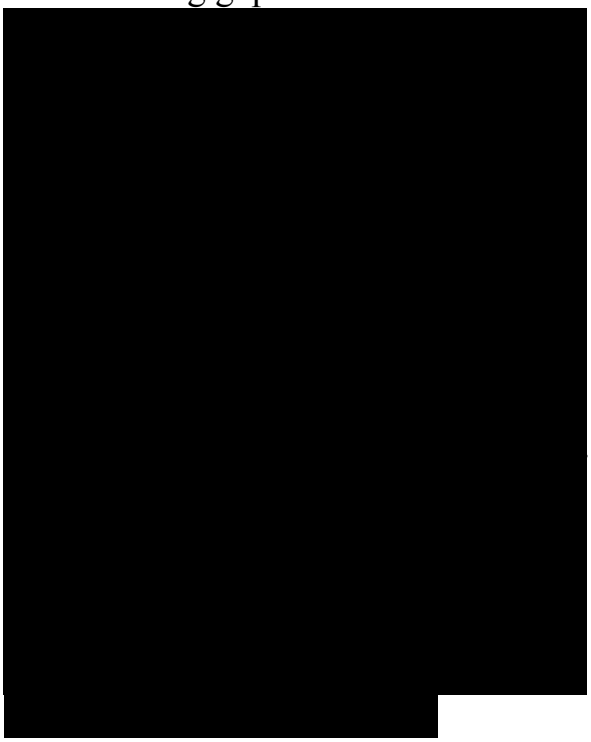
**Dịch tài liệu của bạn:** http://www.mientayvn.com/dich_tieng_anh_chuyen_nghanh.html

| | |
|---|---|
| A Heuristic Approach for Horizontal Fragmentation and Allocation in DOODB | Phương pháp tự tìm tòi để phân mảnh và allocation ngang trong DOODB<br>Allocation: phân bổ, cấp phát, định vị, phân phối |
| Abstract | Tóm tắt |
| Horizontal fragmentation has an important impact in improving the applications performance, that is strongly affected by distributed databases design phase. Although a great effort had been done in this area there still remain lacks to achieve their objective of improving the overall system performance. This paper introduces a new algorithm for horizontal fragmentation of an Object Oriented Distributed Database System OODDBS. The proposed algorithm is based on the idea that addresses vertical fragmentation and allocation simultaneously for relational system but in the context of horizontal fragmentation of an object model. | Phân mảnh ngang đóng một vai trò quan trọng trong việc cải thiện hiệu suất ứng dụng, một yếu tố bị ảnh hưởng mạnh bởi giai đoạn thiết kế cơ sở dữ liệu phân tán. Mặc dù trong lĩnh vực này, người ta đã tiến hành nghiên cứu rất nhiều, nhưng vẫn chưa thể cải thiện được hiệu suất toàn hệ thống như mong đợi. Bài báo này trình bày một thuật toán mới để phân mảnh ngang hệ cơ sở dữ liệu phân tán hướng đối tượng OODDBS. Thuật toán đề xuất dựa trên ý tưởng chỉ định phân mảnh và allocation dọc đồng thời cho hệ thống quan hệ nhưng trong khuôn khổ phân mảnh ngang một mô hình đối tượng. |
| The investigated approach uses a cost model and is globally minimizing the fragmentation and allocation costs. In addition it is easy and accurate implementation of the decision on the constituent groups, as well as it is not only fragmenting class instances into groups, but on the contrary it isalso distributing class instances on sites. | Phương pháp nghiên cứu sử dụng mô hình chi phí và tối thiểu hóa toàn cục phí tổn phân mảnh và allocation. Thêm vào đó, nó dễ dàng và chính xác hóa việc thực thi quyết định về các nhóm thành phần, cũng như nó không chỉ phân mảnh các thể hiện của lớp thành các nhóm, mà trái lại, nó cũng phân bố các thể hiện của lớp trên các vị trí. |
| Validating of the proposed approach is done via a complete simulation. A comparative study proves that the suggested approach has fewer cost and simpler than most recent horizontal | Việc đánh giá phương pháp đề xuất được thực hiện bằng một mô phỏng hoàn chỉnh. Nghiên cứu so sánh chứng tỏ rằng phương pháp đề xuất có giá thành thấp hơn và đơn giản hơn đa số |

| | |
|---|---|
| fragmentation algorithm uses affinities. | thuật toán phân mảnh ngang gần đây sử dụng các phép biến đổi affin.<br>phép biến đổi affin=phép biến đổi đồng dạng |
| 1. Introduction<br>As opposed to centralized databases where the design phase handles only logical and physical data modeling, the design process in Distributed Object Oriented Databases involves as well data partitioning and allocation to the nodes of the system. Horizontal fragmentation, in Object Oriented Database Systems, distributes class instances into fragments. Each object has the same structure and a different state or content. Thus, a horizontal fragment of a class contains a subset of ~~thewhole class extension~~. So, it can be said that fragmentation plays an important role in the design phase of distributed Database.<br>Horizontal fragmentation is usually subdivided in primary and derived fragmentation. Many of the existing Object Oriented (OO) fragmentation approaches are usually inspired from the relational fragmentation techniques. So, it can be said that fragmentation plays an important role in the design phase of Distributed Database. Next section will explores the considerable effort that had been documented in horizontal fragmentation. | 1.Giới thiệu<br>Trái ngược với các cơ sở dữ liệu tập trung trong đó giai đoạn thiết kế chỉ xử lý mô hình dữ liệu logic và vật lý, quá trình thiết kế trong Các Cơ Sở Dữ Liệu Hướng Đối Tượng Phân Tán liên quan đến việc phân vùng dữ liệu và allocation đến các nút của hệ thống. Phân mảnh ngang, trong các hệ cơ sở dữ liệu hướng đối tượng phân tán các thể hiện của lớp thành từng mảnh. Mỗi đối tượng có cấu trúc như nhau và trạng thái hoặc nội dung khác nhau. Vì thế, việc phân mảnh ngang một lớp chứa một tập con toàn bộ phần mở rộng của lớp. Vì vậy, chúng ta có thể nói rằng phân mảnh đóng vai trò quan trọng trong giai đoạn thiết kế cơ sở dữ liệu phân tán. Phân mảnh ngang thường được chia nhỏ thành phân mảnh nguyên thủy và thứ cấp. Nhiều phương pháp phân mảnh hướng đối tượng trong hiện tại (OO) thường có nguồn gốc từ các kỹ thuật phân mảnh quan hệ. Vì vậy, chúng ta có thể nói rằng phân mảnh đóng vai trò quan trọng trong giai đoạn thiết kế Cơ sở dữ liệu phân tán. Trong phần tiếp theo, chúng ta sẽ phân tích những nổ lực đáng chú ý đã được ghi nhận trong phân mảnh ngang. |
| Bellatreche et al. [1] present a horizontal fragmentation method and an analytical cost model to evaluate query execution time in horizontally | Bellatreche và các cộng sự [1] trình bày phương pháp phân mảnh ngang và mô hình phân tích chi phí để đánh giá thời gian thực thi truy vấn trong các cơ |

| | |
|---|---|
| fragmented databases. The fragmentation schema with the best performance according to the cost model is achieved through a hill-climbing algorithm, by selecting a subset of classes for primary horizontal fragmentation. | sở dữ liệu phân mảnh ngang. Phương pháp phân mảnh với hiệu suất tốt nhất theo mô hình chi phí đạt được thông qua thuật toán leo đồi, bằng cách chọn một tập con các lớp để phân mảnh ngang nguyên thủy. |
| The work from Ezeife and Barker [2,3] presents a set of algorithms to the horizontal fragmentation of all classes in a database schema. It takes relationships between classes into account to propose primary and derived horizontal fragmentation. However, this approach works at the instance level, where the class instances already exist in the database to proceed with the fragmentation process. It also assumes a storage structure for the objects in the database class hierarchy in which an instance of a subclass physically contains a pointer to the instance of itssuperclass that is logically part of it. This assumption leads to considering inheritance links in the horizontalfragmentation process. | Công trình của Ezeife và Barker [2,3] trình bày một tập hợp các thuật toán để phân mảnh ngang tất cả các lớp trong một lược đồ cơ sở dữ liệu. Nó tính đến mối quan hệ giữa các lớp để đề xuất cách phân mảnh nguyên thủy và dẫn xuất. Tuy nhiên, phương pháp này làm việc ở mức instance, trong đó các thể hiện của lớp cũng tồn tại trong cơ sở dữ liệu để tiến hành quá trình phân mảnh. Nó cũng giả định một cấu trúc lưu trữ cho các đối tượng trong thứ bậc lớp của cơ sở dữ liệu trong đó về mặt vật lý một instance của lớp con chứa một con trỏ hướng đến một instance ở siêu lớp của nó về mặt logic là một phần của nó. Giả thuyết này dẫn đến việc cần phải xem xét các liên kết kế thừa trong quá trình phân mảnh ngang.<br>instance : thể hiện |
| Savonnet et al. [4] propose a methodology for the horizontal fragmentation of all classes in a database schema. The choice between primary and derived horizontal fragmentation on each class considers its relationships, which are defined by analyzing only the method calls between classes in the schema. The | Savonnet và các cộng sự [4] đã đề xuất một phương pháp để phân mảnh ngang tất cả các lớp trong lược đồ cơ sở dữ liệu. Việc lựa chọn giữa phân mảnh ngang nguyên thủy và dẫn xuất trên mỗi lớp xét đến các mối quan hệ của nó, những mối quan hệ được định nghĩa qua việc phân tích việc gọi các phương thức giữa các lớp trong sơ đồ. |

| | |
|---|---|
| work does not present an algorithm to support the methodology.<br>Baiao et al. [5] proposes a complete fragmentation methodology for OODBMS, which is divided in three phases. First, there is an Analysis Phase to assist distribution designers in defining the most adequate fragmentation algorithm (horizontal, vertical, or both) tobe applied in each class of the database schema. The Analysis Phase also considers the case in which no fragmentation of a class is the best alternative. Second, they present an algorithm to perform Vertical Fragmentation in a class. Finally, the authors present an algorithm to perform Horizontal Fragmentation on the whole class or on a vertical fragment of a class, which may result in mixed fragmentation.<br>In [6], Darabant and Campan propose a new algorithm for horizontal fragmentation in object oriented databases with simple attributes and methods. It relies on an AI nonsupervised clustering method (agglomerative hierarchical method [7]) for partitioning classes into sets of similar instance objects. This algorithm groups objects together by their similarity with respect to a set of user queries with conditions imposed on data. Similarity (dissimilarity) between objects is defined in a vector space model and is computed using different metrics. As a result, they cluster objects that are highly used together by queries. | Công trình không trình bày thuật toán để hỗ trợ cho phương pháp.<br>Baiao và các cộng sự [5] đề xuất một phương pháp phân mảnh hoàn chỉnh cho OODBMS, được chia thành ba giai đoạn. Thứ nhất, có một giai đoạn phân tích để hỗ trợ các nhà thiết kế phân tán trong việc xác định các thuật toán phân mảnh thích hợp nhất (ngang, dọc, hoặc cả hai) được áp dụng trong mỗi lớp của lược đồ cơ sở dữ liệu. Giai đoạn phân tích cũng xét trường hợp không có phân mảnh nào của một lớp là lựa chọn tốt nhất. Thứ hai, họ trình bày một thuật toán để thực hiện Phân Mảnh Dọc trong một lớp. Cuối cùng, các tác giả cũng trình bày một thuật toán để thực hiện phân mảnh ngang trên toàn bộ lớp hoặc trên một mảnh dọc của một lớp, điều này có thể dẫn đến sự phân mảnh hỗn hợp.<br>Trong [6], Darabant và Campan đề xuất một thuật toán mới để phân mảnh ngang trong các cơ sở dữ liệu hướng đối tượng với các thuộc tính và phương pháp đơn giản. Nó lệ thuộc vào phương pháp phân nhóm không giám sát AI (phương pháp thứ bậc kết tụ [7]) để phân vùng các lớp thành các tập hợp đối tượng instance giống nhau. Thuật toán này gộp các đối tượng với nhau theo sự tương tự của chúng đối với tập hợp truy vấn người dùng với các điều kiện được áp đặt trên dữ liệu. Sự giống (hoặc khác nhau) giữa các đối tượng được định nghĩa trong một mô hình không gian vector và được tính toán bằng các chuẩn khác nhau. Do đó, chúng gộp các đối tượng được sử dụng nhiều với |

| | |
|---|---|
| Darabantin [8] reporteda new algorithmfor horizontal fragmentation in object-oriented databases with complex attributes and methods. Fragmentation in complex OO hierarchies is usually performed in two steps: primary fragmentation and derived fragmentation. Primary fragmentation groups class instances according to a set of class conditions [9] imposed on their simple attributes. They proposed algorithm unifies the two fragmentation steps: primary and derived into a single step. Each object is represented as a vector and use the k-means clustering algorithm for separating clusters (fragments) of objects. | nhau qua các truy vấn.<br>Darabantin [8] đã trình bày một thuật toán mới để phân mảnh ngang trong các cơ sở dữ liệu hướng đối tượng với các thuộc tính và phương pháp phức tạp. Việc phân mảnh trong các hệ thống phân cấp OO phức tạp thường được thực hiện theo hai bước: phân mảnh nguyên thủy và phân mảnh dẫn xuất. Phân mảnh nguyên thủy gộp các thể hiện của lớp theo một tập hợp các điều kiện lớp [9] được áp đặt trên các thuộc tính đơn giản của chúng. Họ đề xuất thuật toán hợp nhất hai bước phân mảnh: nguyên thủy và dẫn xuất thành một bước duy nhất. Mỗi đối tượng được biểu diễn dưới dạng một vector và sử dựng thuật toán phân cụm k-trung bình cho các cụm đối tượng riêng biệt (các mảnh) |
| 1.1 Contributions<br>The new suggested approach for horizontal fragmentation is more simpler, easier and accurate than any other algorithms using affinities, mentioned in literatures[1,2,3,4,5,6,8]. This approach is based on the relational model approach introduced by Scheme[10] for vertical fragmentation process which is adapted for horizontal fragmentation of object oriented database. By this algorithm , the two main processes of object oriented database design( fragmentation and allocation) are done simultaneously with no kind of complexity and with the smallest allocation cost comapred with other ~~horizontal fragmentation algorithms~~ | 1.1 Các đóng góp<br>  |

~~using affinities.~~

The paper is organized as follows : the next Section summarize the main needed definitions .The suggested heuristic approach for horizontal fragmentation of object oriented database is presented in Section 3. While section 4 evaluates and compares the new suggested approach with an old one in literature to show the effectiveness of the new approach. Finally, Section 5 concludes this paper.

2. Basic definitions

Some definitions are needed in the presence of the class fragmentation algorithms which will be introduced next.Let Q={q1,q2,……..,qq) be the set of user queries(applications) running methods from the set of methods denoted by{Mi1.j,Mi2.k,……. Min.p}.The cardinality of the class is the class instance object number.(denoted card(Ci))]. The following definitionsare from [2,11,12].

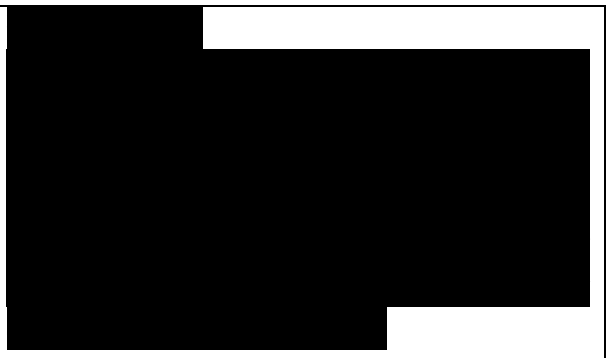
Definition 1

Given a set of queries Q={q1,q2,….qq} that will run on a class C(A,M) . Each query qj( 1□ j □ q) has the following parameter:

This generates the insatance usage matrix(IUM),whose rows are the queries, and the columns are the instances.

Definition 2

The predicate affinity between two predicates pi, and pj of a class C(A,M) with respect a set of queries Q={q1,q2,….qq} concerns the predicate of class C(A,M) and predicate of the owner class(es).this affinity measure is defined as follows:



where:

I1= all C queries that access pi, and pj of the class C.

I2= all owner C queries that access pi, and pj of the class of C.

All affinity measures are grouped in a matrix called the predicate affinity matrix(PAM).

3. The proposed heuristic approach for fragmentation and allocation

This section presents the proposed horizontal fragmentation approach which based on the approach suggested for relational database [10] and applied it on class models consisting of simple attributes using simple methods. The horizontal fragmentation algorithm function is to break class instances (rows) into a set of smaller classes (fragments) that make used applications able to execute using only one fragment [8]. The main steps of the suggested approch are summarized below.

Input:

• Qn={Q1,…..,Qn} /* set of queries for class C.

• C={i1,i2,…., ii ,p1,p2,…., pp) /* a

class with a set of instances,i is the number of instances , and a set of predicates used to access its attributes, p is the number of predicates.

• N={1,….,k) /* a set of network nodes.

Main Steps:

1) Calculate the request at each site for each instance to construct anInstance request Matrix (IRM).Each instanceis requested by multiple queriesfrom one site. The request of a instance at site h is the sum of frequencies of all queries at the site accessing this attribute it can be calculated with the equation below:

where:

Ii is the required method, h is the site needingthis instance Ii , Fji is the frequency of a query accessing a instance Ii, m is the total number of predicateson the site h.

2) Calculate the pay at each site for each instance to construct the Instancepay Matrix.

The term pay measures the costs of allocating a single instance to a network node . The allocating pay of a instance Ii to a site h measures the costs of accessing method mi from all queries at other sites h' ,which is different from h. To calculate the pay parameter use the following equation :

where :

chh' is the transmission cost between the two sites h and h' ,Note that cost factor chh'=0 if h=h'

3) Cluster each instanceto the site which has the lowest value ofthe pay .

Output:

Horizontal fragmentation schema(class fragments) and fragment allocation schema(the allocated site for each fragment). This approach is firstly finds the site that has the smallest pay value, and then allocates the instance to the site. So, horizontal fragmentation and allocation schema are obtained in the same step.

4. Exprimental Results and Evaluation

Simulation example for a three sites is presented for making a comparison between the applying of Navath's algorithm and the new suggested approach on a set of data explained in details later.

4.1 DataBase schema

An illustrative example[11] that will be used in the evaluation process , will be demonstrated andsome queries for the database classes are presentedin table 1. The used example consists of six classes; each class consists of its own simple attributes and simple methods which will be distributed over three nodes (sites). Database schema shown in figure 1.

4.2 Preparation step

The common used matrixbetween Navath'sand the new suggested approach is Application Frequency Matrix (AFM) for the queries of each case. This matrix illustates the number of requirements of each query by each site of the three available sites .Also the Transportation Cost Factorsare given in table 2. All submitted queries for

Table1 All submitted queries for Person class

Person class are listed in table1 . By these queries , 40 case studies are constructed to be used in the evolution process and in the comparative study between the Navath's algorithm and new suggested approach for horizontal fragmentation of object oriented databases,these cases are shown in table 3.

1.1 Applying navath's horizontal algorithm

In this part of evaluation the Navath's algorithm which has four steps, is applied on the Person class. It firstly, applies the Bond Energy Algorithm (BEA) and then applies the Binary Vertical Partitioning (BVP) Algorithm which divides the class methods into two fragments only. The algorithm steps [2] , are as follows:-

●Step1:

Constructing the predicate usage matrix (PUM) by using definition 1 mentioned ealier.

●Step2:

Constructing the predicate affinity matrix (PAM) which is drived from the given PUM,use definition 2.

●Step3:

Applying the Bond Energy Algorithm (BEA) which tries to reorder columns and rows of the constructed PAM to form a new matrix that called "clusted affinity matrix" (CAM) . BE has two steps:

Initialization:

Place and fix one of the columns of PAM into CAM

Iteration Step:

Pick each of the remaining n-i columns (where i is the columns alredy placed in CAM) and try to place them in the remaining i+1 positions in the CAM matrx.Coose the placement that makes the greatest contribution to the global affinity measure .continue untill no more columns remain to be placed.

Notice:

1 - The bond between two predicates i and j is given by the equation given as:

where :

Affinity is calculated by definition 2 which mentioned before.

2 - The net contribution to the net global affinity measure of placing the method k between i and j is : contikj □ 2 * bond ik 2 * bond kj 2 * bond ij

Row Reordering Step:

Once the column ordering is determined ,the placement of the rows should also be changed so that their relative positions match the relative positions of columns [13].

●Step4:

Fragmenting the class predicates by using the Binary horizontal Partitioning Algorithm which fragmrnt these predicates into only two non overlapping fragments(upper and lower fragment) according to the following equation [4]:-

where:-

- CL and CU : counts the total number of transactions accesses that need only one fragment
- CI : counts the total number of transactions accesses that need both fragments.

●Step5:

Calculating the allocation cost of each fragment(instances) using the transportation cost factores in table2.

4.4 Applying the new suggested approach

The main steps of new suggested approach are three which are mentioned before in section ш . These steps are:.-

●Step1:

Constructing the instance request matrix(IRM),which calculates the request at each site for each instance (use equation 1).

●Step2:

Constructing instance pay matrix (IPM) ,which calculates the pay at each site

for each instance(using equation 2).

●Step3:

This step Cluster each instance to the site which has the lowest pay value.

4.5 Discussion

By comparing the total cost , from figure 2 , for each case of the 40 cases for Navath's algorithm and new suggested approach , it noticed that the new approach is better than the Navath's s because it often reduces the allocation cost and not only that but also doing allocation process for each class instance simultaneously in the same time with fragmentation process. From the above discussion, it is clear that the suggested heuristic approach for horizontal fragmentation gives better results than any other horizontal fragmentation approaches because it gives the decision according to the affinities between two instances based on the cost at each site.

5 Conclusion

Horizontal fragmentation is one of the algorithms that used for the distributed object oriented database design. This paper is interested in finding a new simple and easier horizontal fragmentation approach. The new suggested heuristic horizontal fragmentation approach has three advantages. Firstly, is less complex than any other horizontal fragmentation algorithm using instance affinities .Secondly, gives the decision of choosing the fragmentation schema depending on the minimum cost of allocation of each instance between the

| | |
|---|---|
| available sites. Thirdly, not doing only fragmentation but also doing allocation simultaneously in the same step. | [redacted] |
| Fig.2 The Cost by using Navath's Algorithm and New Approach for each Case | [redacted] |