International Journal of Engineering Research and Modern Education (IJERME)
ISSN (Online): 2455 - 4200
(www.rdmodernresearch.com) Volume I, Issue I, 2016

HOSPITAL MANAGEMENT: DATA RETRIEVE FROM LARGE DATA SET BY USING DYNAMIC SEARCH ALGORITHM

Leihaorungbam Bikash Singh* & K. Adlin Suji**
* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu
** Associate Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

Efficient top-N retrieval of records from a database has been an active research field for many years. We approach the problem from a real-world application point of view, in which the order of records according to some similarity function on an attribute is not unique. Many records have same values in several attributes and thus their ranking in those attributes is arbitrary (based on random choice). For instance, in large person databases many individuals have the same first name, the same date of birth, or live in the same city. Existing algorithms are ill-equipped to handle such cases efficiently. We introduce a Dynamic TMS Searcher, which retrieves larger chunks of records from the sorted lists using fixed limits, and which focuses its efforts on records that are ranked high in more than one ordering and are thus more promising candidates. We experimentally show that our method outperforms Dynamic Sorting Algorithm (DSA) for top-k retrieval in those very common cases where we used with dynamically scheduling the resources based on the data which are provided with, this efficient short search algorithm along with the massive data retrieval on a very fine tuple data's can be of a different dataset. Here in this paper we are going to use these logics for the need of solution in the field of medical research. Where there are many manageable databases that are been used in a common path for the end of healthy need and the retrieval of solution for the cause of illness to a human being.

Key Words: DSA Algorithm, Massive Data, Table Scan & Selective Retrieval

1. Introduction:

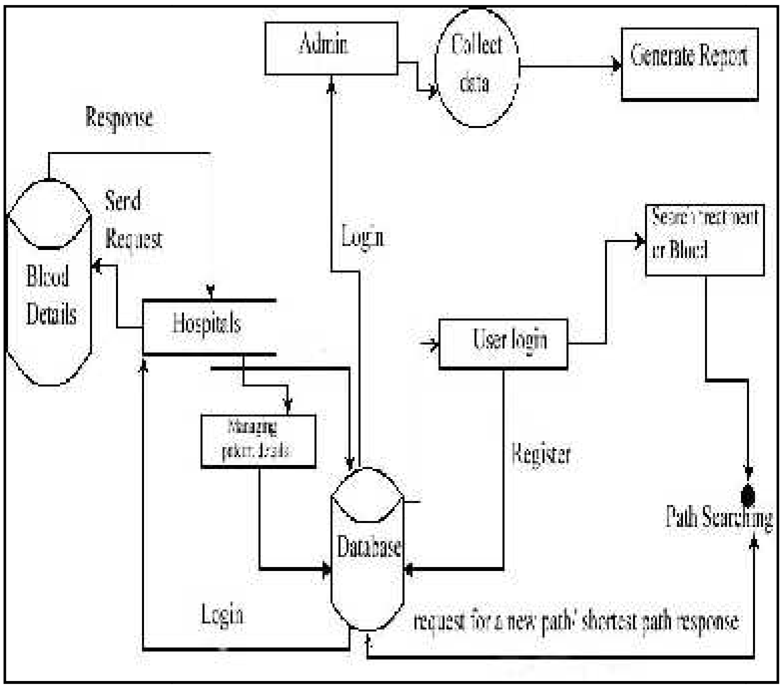


Figure 1: Architecture

The scope of this project is to take solution for diseases which affecting human being, after registering into the application, details of disease affecting human being will be updated and solution would generate to prevent the disease occur again to another people. In this paper, we have proposed an efficient algorithm called dynamic sorting algorithm (DSA).Finder to find the optimal solution, whose success lies in a set of effective pruning strategies and a novel index structure. Extensive experiments on two real-life datasets demonstrate the efficiency and effectiveness of our solution.

Since the whole discovery process with straightforward solutions can be very lengthy in a practical dataset, we propose a series of techniques which address the Indexing, searching and updating issues respectively.

The contributions of this paper are as follows:

- ✓ This paper proposed Dynamic Sorting Algorithm to search and retrieve data very easily from huge databases.
- ✓ The top results only shown when we search any kind of data
- ✓ An early termination will helps to terminate the repeated terms in the table and we can save the data in an efficient manner in the database.
- ✓ The experimental results show that DSA has a significant advantage over the existing algorithms.

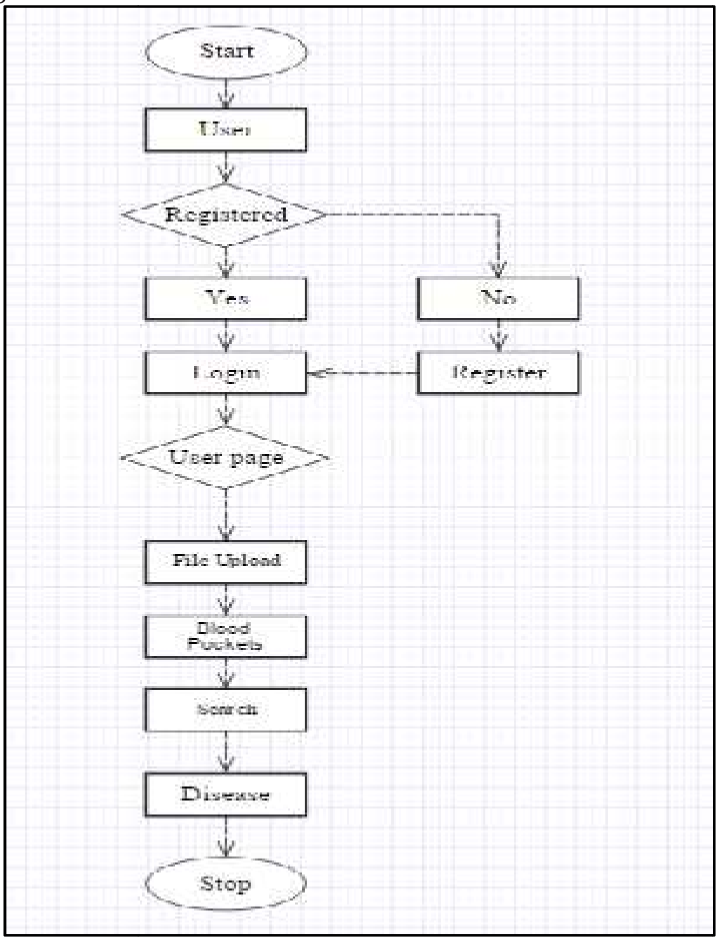


Figure 2: System Flow Diagram

The rest of the paper is organized as follows. Section 2 reviews related work. Existing system are described in Section 3. Proposed systems are described in section 4. DSA algorithm is introduced in Section 5 (the feasibility report), Section 6 Results are in section 7 (conclusion) in section 8 and future enhancement will be the following section.

2. Related Work:

2.1 Selection Sort Algorithm:

The selection sort improves on the bubble sort by making only one exchange for every pass through the list. In order to do this, a selection sort looks for the largest value as it makes a pass and, after completing the pass, places it in the proper location. As with a bubble sort, after the first pass, the largest item is in the correct place. After the second pass, the next largest is in place. This process continues and requires n−1n−1 passes to sort *n* items, since the final item must be in place after the $(n-1)^{st}$ pass.

The selection sort makes the same number of comparisons as the bubble sort and is therefore also O(n2)O(n2). However, due to the reduction in the number of exchanges, the selection sort typically executes faster in benchmark studies. In fact, for our list, the bubble sort makes 20 exchanges, while the selection sort makes only 8.

2.2 Find Index of Smallest:

Here we have done about to find the data to retrieve from massive data set. So, data is massive then we should find through index i.e. searching. Every index has some data which easy to find the data.

2.3 Selective Retrieval:

Ability to remember some facts while apparently forgetting others, especially, when they are inconvenient, retrieve certain facts and events but not others.

3. Existing System:

- ✓ The existing application provides query based table scan to retrieve the results.
- ✓ They also use indexes with specific attributes to build the performance on view.
- ✓ It also has bound based fashion which will consist of lower-bound and upper-bound scores.
- ✓ The cost of pre-computing the data structures and update process is a major cost.
- ✓ Database of all hospital cannot be maintained by government
- ✓ Search efficiency is slow.
- ✓ Survey is taken manually at the end of the year and it has human errors.

3.1 Advantages of Existing System:

- ✓ In this system we can able to prevent the disease to occur again.
- ✓ We can able to generate medicines for the diseases.
- ✓ We can get the details of blood donors under one roof.

3.2 Disadvantage of Existing System:

- ✓ We cannot manage the whole record of diseases manually.
- ✓ It's difficult take solution on time.
- ✓ We cannot search the blood donor in a short time and details of them are not available to all.

4. Proposed System:

- ✓ In the proposed system we use Dynamic Sorting Algorithm to search and retrieve data very easily from huge databases.
- ✓ It reduces time complexity.
- ✓ Efficiency is very high.
- ✓ The top results only shown when we search any kind of data.
- ✓ And the database of the each hospital is maintained by the government.
- ✓ This database will helps to identify the patient affected by which diseases
- ✓ And this information will help to take the remedy of the government

- ✓ An early termination will helps to terminate the repeated terms in the table and we can save the data in an efficient manner in the database.

4.1 Advantage of Proposed System:

- ✓ We can able to prevent the disease in a short time.
- ✓ We can save the time for the searching process.
- ✓ We can easily get the details of blood donors.

5. DSA Algorithm:

The nature of the search problem can be big in terms of the massive data search to make the efficient results provided for the end user. Each time when the auxiliary index becomes too large, we merge it into the main index. The cost of this merging operation depends on the resource needed and sometimes there may be resource engaged with another process this makes the result to be delay.

Indexing Based on:

Primary Key: single attribute, no duplicates.

Secondary Keys: one or more attributes ‰duplicates are allowed ‰indexing in M-dimensional feature spaces.

5.1 Selection Sort Search Pseudo Code:

```
Selection Sort(A)
// GOAL: place the elements of A in ascending order
n := length[A]
for i := 1 to n
// GOAL: place the correct number in A[i]
 j := FindIndexOfSmallest( A, i, n )
swap A[i] with A[j]
// L.I. A[1..i] the i smallest numbers sorted
end-for
end-procedure
FindIndexOfSmallest( A, i, n )
// GOAL: return j in the range [i,n] such
//that A[j]<=A[k] for all k in range [i,n]
smallestAt := i ;
for j := (i+1) to n
if ( A[j] < A[smallestAt] ) smallestAt := j
// L.I. A[smallestAt] smallest among A[i..j]
end-for
return smallestAt
end-procedure
```

Counting Code:

- ✓ In Find Index of Smallest, dominated by line 3, run n-i times.
- ✓ In Selection Sort, dominated by line 4, run n times, the i^{th} call to Find Index of Smallest taking time n-i, i=1, ..., n.

6. Feasibility Study:

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- ✓ Technical Feasibility

- ✓ Operation Feasibility
- ✓ Economic Feasibility

6.1 Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ✓ Does the necessary technology exist to do what is suggested?
- ✓ Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- ✓ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- ✓ Can the system be upgraded if developed?
- ✓ Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a browser based user interface for audit workflow. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

The software and hard requirements for the development of this project are not many and are already available or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

6.2 Operational Feasibility:

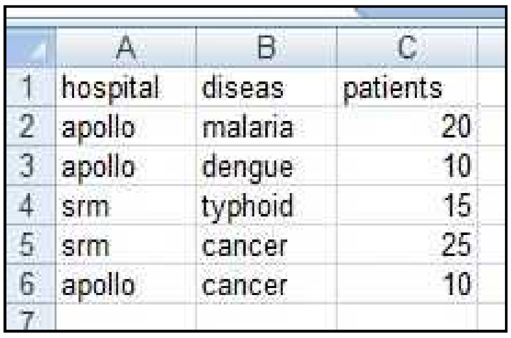
The analyst considers the extent the proposed system will fulfill his departments. That is whether the proposed system covers all aspects of the working system and whether it has considerable improvements. We have found that the proposed "Secure transaction" will certainly have considerable improvements over the existing system.

6.3 Economic Feasibility:

The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approachable. Working in this system need not required a highly qualified professional. The operating-environment costs are marginal. The less time involved also helped in its economic feasibility.

7. Results:

The results will be shown as tables in Microsoft excel and automatically save to our hard-drive. The figure 3 is one of the screen shot of that which describe hospital names, number of patients and disease.



	A	B	C
1	hospital	diseas	patients
2	apollo	malaria	20
3	apollo	dengue	10
4	srm	typhoid	15
5	srm	cancer	25
6	apollo	cancer	10
7			

Figure 3: Report of Patient

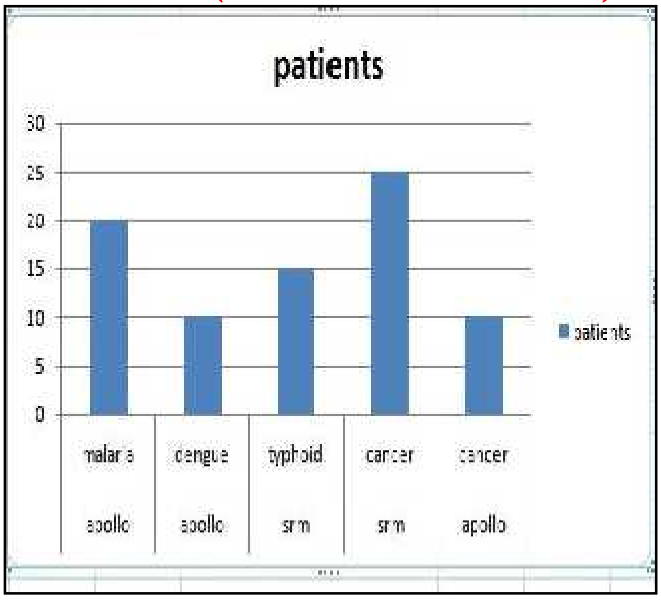


Figure 4: Patient Chart

The above figure 4 shows the column charts of the hospital reports by the admin. It helps to describe how much people who suffering from which diseases and which area where affected most.

8. Conclusion:

Efficient top-N retrieval of records from a database has been an active research field for many years. We approach the problem from a real-world application point of view, in which the order of records according to some similarity function on an attribute is not unique. We experimentally show that our method outperforms Dynamic Sorting Algorithm (DSA) for top-k retrieval in those very common cases where we used with dynamically scheduling the resources based on the data which are provided with. Here in this Project we are going to use these logics for the need of solution in the field of medical research, to prevent people dying for same issue and to provide solution for the disease in large scale.

9. Future Enhancement:

Optimization of this application, we have some additional features to deliver by more attractive to the user experience. Our future enhancement is to provide prescription and giving suggestions to the patients through online. So that they can easily clarify their doubts and get help for emergency cases.

10. References:

1. Xixian Han, Jianzhong Li, Member, "Efficient Top-k Retrieval on Massive Data", IEEE Transactions on Knowledge of data engineering, 2015, vol.27 (10), 2687-2699.
2. Prabhudev. S. Irabashetti, "Dynamic Search Algorithm in P2P Networks", in Proceedings on International Journal of Innovations in Engineering and Technology (IJIET), December 2013, vol. 3(2), 64-75.
3. Ihab F. Ilyas, George Beskales and Mohamed A. Soliman David R," A Survey of Top-k Query Processing Techniques in Relational Database Systems", in proceedings ACM Journal Name, Vol. V, No. N, Month 20YY, 1-61.
4. Dehua Chen1, Changgan Shen2, Jieying Feng3 and Jiajin Le," An Efficient Parallel Top-k Similarity Join for Massive Multidimensional Data Using Spark", in Proceedings International Journal of Database Theory and Application, 2015, vol.8(3),pp 57-68.

5. Hsinping Wang, Tsungnan Lin, Chia Hung Chen, and Yennan Shen.," Dynamic Search in Peer-to-Peer Networks" in Processing: NSC grants NSC92-2213-E-002-087, pp 332-333.
6. R. Akbarinia, E. Pacitti, and P. Valduriez, "Best position algorithms for top-k queries," in Proc. 33rd Int. Conf. Very Large Data Bases, 2007, pp. 495–506.
7. H. Bast, D. Majumdar, R. Schenkel, M. Theobald, and G. Weikum, "Io-top-k: Index-access optimized top-k query processing," in Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 475–486.
8. Y. Chang, L. Bergman, V. Castelli, C. Li, M. Lo, and J. Smith, "The onion technique: Indexing for linear optimization queries," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp. 391–402.
9. G. Das, D. Gunopulos, N. Koudas, and D. Tsirogiannis, "Answering top-k queries using views." in Proc. 32nd Int. Conf.Very Large Data Bases., pp. 451–462, 2006.
10. R. Fagin, R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2003, pp. 301–312.
11. R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," in Proc. 20th ACM SIGMOD-SIGACTSIGART Symp. Principles Database Syst., 2001, pp. 102–113.
12. R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," J. Comput. Syst. Sci., vol. 66, no. 4,pp. 614–656, 2003.