International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 15470-15476

Website Restructuring using Fibonacci Heap and Split Based Frequency Count

K. Shyamala[1], S. Kalaivani[2]

[1]Associate Professor, [2]Research Scholar, PG & Research Department of Computer Science, Dr.Ambedkar Government Arts College (Autonomous), Affiliated to University of Madras, Chennai, India.

E-mail: [1]shyamalakannan2000@gmail.com; [2]kalai5391@gmail.com

Abstract

Web usage mining is used to mine valuable information from a web server log file. Web users' interest changes over time and it is not stable. Hence, the static website will get out-dated. So website needs to be modified with minimal changes to meet user requirements. The main intention of this research is to retrieve require web pages with minimal search cost in a website. In this work, Split Based Frequency Count (SBFC) algorithm is proposed to find frequently accessed web pages from pre-processed web server log file. After finding the frequently accessed web pages, Fibonacci heap tree is constructed based on the frequency count. Three different types of data structures, namely Binary Search tree, Max heap and Fibonacci heap are used for reorganization. Experimental results show that Fibonacci Max heap tree gives better performance and minimal search cost between the web pages. It is suitable for the dynamic website which needs a change periodically.

Keywords: Web usage mining, Binary search tree, Max heap, Fibonacci max heap.

1. INTRODUCTION

Website navigation has become one of the most significant design features in many domains, including finance, entertainment, e-commerce, government, medical and education. Navigation through a large website for finding relevant information can be tedious and frustrating. The act of facilitating the reconstitution of pages is known as personalization. In another case, modifying the website structure will facilitate the navigation for users and it is known as web transformation. Web usage mining, web content mining and web structure mining are the three categories of web mining. Web usage mining is used to identify user behavior and web user activity from web server log file [1]. Web server log file stored on the web server contains the interaction between the web user and website. Web user navigation and user behavior can be identified by analyzing the weblog [2].

Tree is the most suitable data structure for storing and retrieving data efficiently from memory. In the meanwhile, this could be authentic only when the tree is height balanced. Binary search tree [3] is a non linear data structure that can be defined as the node based data structure.

The node consists of root and two subtrees which include left subtree and right subtree. In general situation we look forward for the tree with minimal height. It is feasible only when the tree is height balanced with node.

If the keys are included in random order then a binary search tree depends upon 1.36 (log(n)) comparison. In binary search tree each node x stores a key value, such that the left subtree of a node with lesser key value than the root node and then the right subtree of a node with greater key value than the root node. While doing search operation, binary search tree property will be very much useful. The search operation on a binary search tree is proportional to the height of the binary search tree and the worst case time complexity for a complete binary tree with node n is O(log(n)). The number of links from the root to child node is defined as the height of the binary search tree.

1.1 Fibonacci Max Heap Tree

Fibonacci heap is an accumulation of minimum-heap trees. Every tree root node in the collection has a property that, the key value of a child should be greater or equal to the key value of the parent. These properties imply that the smaller key is always remains at the root of the tree. All tree roots are connected usingthe circular doubly linked list, so a single pointer can access the minimum value easily. The amortized time complexity of the insertion and find-min are all O (1) as in the case of the Fibonacci heap structure.

In this paper, we considered a Maximum element based Fibonacci heap, which is a minor variation on the Fibonacci Min-Heap. Here, instead of pointer points toa minimum element, the pointer points to the maximum element. The insertion and Find-Max also gave the time complexity of O(1) only. This property is very much helpful to organize the web pages.

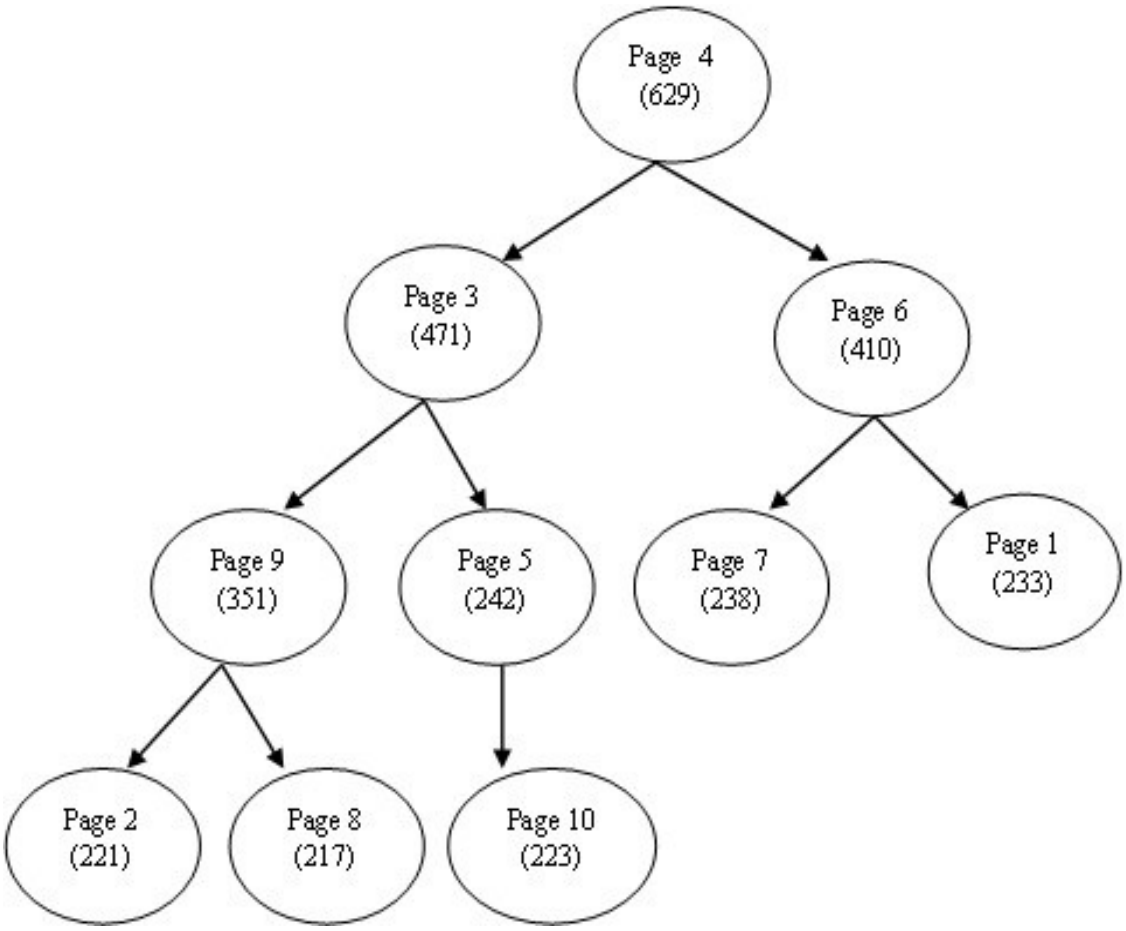

Fig. 5. Max Heap organization of web pages.

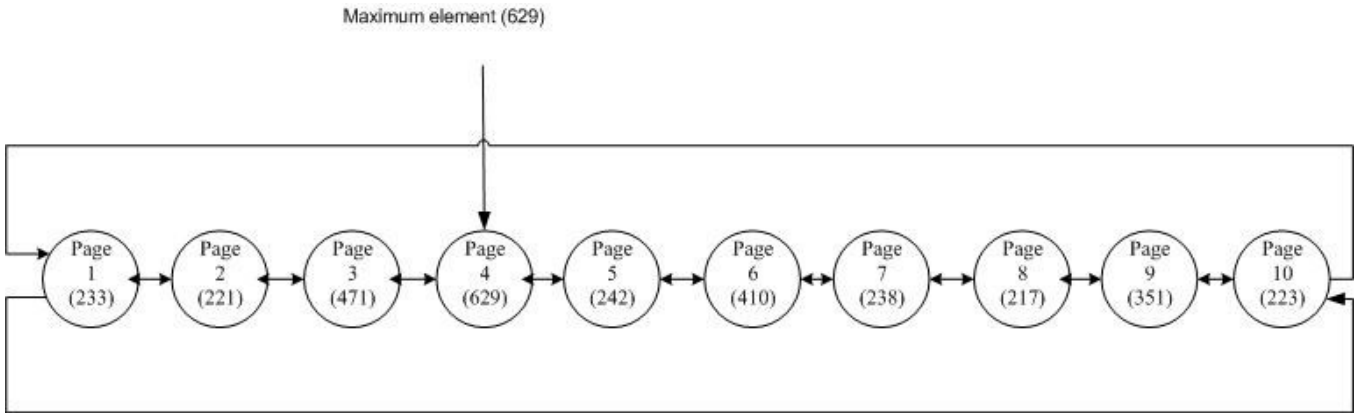

Fig. 6. Fibonacci Maxheap organization of web pages.

The website is well organized with interesting and useful pages for the users [18], which are nearer to the root node. It helps to improve the distance between web pages.

Table 2. Search Cost Calculation.

Page No	Frequency Count	Binary Search Tree		Binary Max Heap Tree		Fibonacci Max Heap Tree	
		Frequency	Search Cost	Frequency	Search Cost	Frequency	Search Cost
Page1	233	233 * 1	233	233 * 3	669	233 * 1	233
Page 2	221	221 * 2	442	221 * 4	884	221 * 1	221
Page 3	471	471 * 2	942	471 * 2	942	471 * 1	471
Page 4	629	629 * 3	1887	629 * 1	629	629 * 1	629
Page 5	242	242 * 3	726	242 * 3	726	242 * 1	242
Page 6	410	410 * 4	1640	410 * 2	820	410 * 1	410
Page 7	238	238 * 4	952	238 * 3	714	238 * 1	238
Page 8	217	217 * 3	651	217 * 4	868	217 * 1	217
Page 9	351	351 * 5	1755	351 * 3	1053	351 * 1	351
Page 10	223	223 * 3	669	223 * 4	892	223 * 1	223
		Total	**9897**		**8197**		**3235**

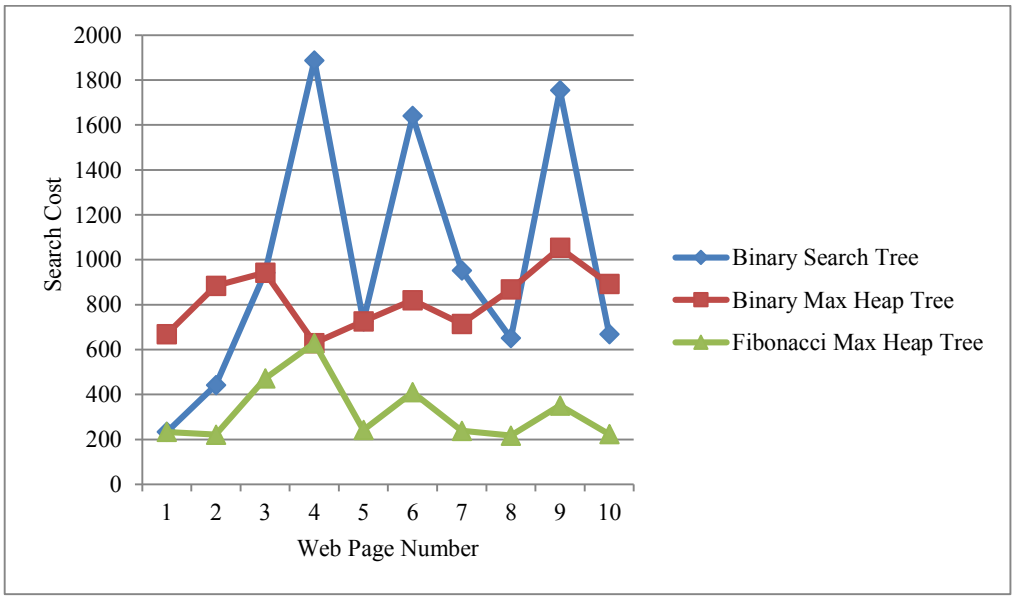

Fig. 7. Graphical representation of three organizations.

The Fig.7. shows the search cost of the three methods considered in this paper. Clearly the diagram shows that the Fibonacci Max heap gives a lowest total search cost of the frequently accessed web pages.

The search cost for each page before and after the reorganization of the website is made and represented in the fig. 7. Experimental results show that the Fibonacci Max heap operation is suggestive to dynamic websites. Dynamic websites are those websites that need to be reorganized periodically based upon the season. Few examples of dynamic websites are e-shopping, e-commerce, educational institutions etc.Users are likely to access the web pages that are relevant to that particular season. The proposed work on the reorganization of the website based on the Fibonacci Max heap structure to bring frequently accessed web pages nearer to the root will minimize the searching time.

8 CONCLUSION

Recommendation algorithms suggest web pages to users, based on their current visit and past user navigation patterns from a web server log file. In this paper, website reorganization is discussed based on the frequently accessed web pages from the pre-processed web server log file. An algorithmSplit Based Frequency Count (SBFC)is proposed to find frequently accessed web pages and it is implemented in Java Net beans 8.2. Instead of considering Fibonacci Min Heap property, we have considered Fibonacci Max Heap property which gives better performance for website reorganization. Web Pages were reorganized based on the frequency count of each page.After the reorganization of web pages, the search cost is reduced. Search cost is calculated based on the Binary Search Tree, Max Heap and Fibonacci Max Heap tree. As a result of the experiments, Fibonacci Max Heap gives better performance. After the reorganization, the website facilitates the web pages which are frequently accessed web pages.

REFERENCES

[1] Kalaivani, S., Shyamala, K.: A Novel Technique to Pre-Process Web Log Data Using SQL Server Management Studio. International Journal of Advanced Engineering, Management and Science. pp.2454--1311(2016)

[2] Kalaivani, S., Shyamala, K.: Clustering of Web users Behavior based on the Session Identification through Web Server Log File, International Journal of Control Theory and Applications.pp.7--16(2017)

[3] Michael L.Fredman., Robert EndreTarjan.: Fibonacci Heaps and their uses in improved network optimization Algorithm, Journal of the Association for Computing Machinery. pp.596--615(1987)

[4] Chang-Chun Lin.: Optimal Website reorganization considering information overload and search depth,European Journal of Operational Research. pp.839--848(2006)

[5] Mohan Krishna, D.V., Yedukondalu, N., Arun Kumar Reddy, D.:Effective User Navigability through Website Structure Reorganizing Using Mathematical Programming Model.International Journal of Computer Trends and Technology (IJCTT).pp.128--135 (2014)

[6] Christopoulou, E., Gorofalakis, J.,Markris, C., Panagis, Y., Sakkopoulos. E., Tssakalidis, A.: Techniques and metrics for improving website structure.J.Web Eng.pp.90--104(2003)

[7] Christos Makris, YannisPanagis, EvangelosSakkopoulos and AthanasiosTsakalidis.:An Algorithmic Framework for Adaptive Web Content.Studies in Computational.pp.1--10 (2006)

[8] Shyamala, K., .Kalaivani, S.: An Effective Web page Reorganization through Heap Tree and Farthest First Clustering Approach. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017). IEEE-CATALOG NUMBER: 978-1-5386-0814-5. (2017)

[9] Suguna, R., and Sharmila, D.: An Overview of Web Usage Mining.International Journal of Computer Applications.(2012)

[10] Joshila Grace, L.K., Maheswari.:DhinaharanNagamalai.,: Analysis of Web Logs and Web User in Web Mining. International Journal of Network Security & Its Applications (IJNSA).(2011)

[11] NehaGoel., Sonia Gupta., C.K. Jha.:Analyzing Web Logs of an Astrological Website Using Key Influencers. International Research Journal.(2015)

[12] Yew ChuanOng.,ZurainiIsmai.: Enhanced Web Log Cleaning Algorithm for Web Intrusion Detection. Recent Advances in Information and Communication Technology Advances in Intelligent Systems and Computing. pp.315--324(2014)

[13] Ankit, R Kharwar.,Chandni A Naik.:Niyanta K Desai.: A Complete Pre Processing Method for Web Usage Mining.International Journal of Emerging Technology and Advanced Engineering. pp.638--641(2013)

[14] Sumathi, C.P.,PadmajaValli, R.,Santhanam, T.: An Overview of Pre-Processing of Web Log Files for Web Usage Mining.Journal of Theoretical and Applied Information Technology.(2011)

[15] Dipa Dixit., Kiruthika, M.:Pre-Processing of Web Logs.International Journal of Computer Science and Engineering. pp.2447--2452(2010)

[16] U.S Govt website: https://www.sec.gov/dera/data/edgar-log-file-data-set.html

[17] Muthusundari, S.,Suresh, R.M.: A Sorting based Algorithm for the Construction of Balanced Search Tree Automatically for smaller elements and with minimum of one Rotation for Greater Elements from BST. International Journal of Computer Science and Engineering (IJCES).pp.297--303(2013)

[18] Thulase, M.B.,Raju,G.T.: Website Reorganization for effective latency reduction through splay and heap tree structures. International Journal of Computer Engineering & Technology (IJCET). (2012) pp.487--498