**IF GOOGLE ANALYTICS WAS RUN BY AMATUERS**

System is a reference to the program as a whole: Outer list refers to the list that contains the lists within it. Inner list refers to the lists that contain the actual information from the data collected.

Storage:

I would use a 2D list to store the information. Each node contains another list and a pointer to the next data received. Each of the lists within the nodes contain all the data from one use of a web page. The nodes would each be one specific piece of information (URL, time visited, operating system). Anything that could be turned into a string or int would be to save checking time. Every time the system received a new piece of information it would add another node to the outer list. This is superior to array storage because size is not an issue. The list “table” never has to resize itself.

Query execution:

When executing a query, the system would create a new list to **hold** the requested information (infoList). It would also create an object that contained the type of information requested (ex. user, time, date) and that contained a new list (the checker list) that was the same size as each inner list. However instead of containing one piece of information, the checker list would contain as many as inputted. For example, if the owner wanted all the users from America OR Canada, the system would make a node that contained both pieces of information. If the user didn’t specify any qualification (ex. They didn’t care about the time aspect) the system would still create a node, but would insert a null object instead of the piece of info. That way the checker list and the actual lists of the table would remain the same size and would allow for faster checking.

After building the checker list the system would go through each inner list and compare the nodes of the checker list to those of the inner list (the checker list and actual list of the table are parallel in construction). The logic would go something like if(checkerNode.getInfo() == null || node.getInfo().equals(checkNode.getInfo())… The checker list would go through each node of the inner list and would break if they didn’t match up.

For things that can’t just be compared like date constraints (between Jan 1 and Feb 1), the system would have a within() method to see if the info from the list was within specifications.

Additionally, the checker list would keep a tracker so it would know when it got the requested information. It would store the info in a temporary variable. If the list matched all the criteria it would add the information to the infoList. Otherwise, the variable would simply get eaten by the garbage collector.

Run Time:

n = the number of specific pieces of info received.

The run time would be O(n) in terms of data visited, not comparison. It would go to each piece of info one time.