

An Evaluation of the Critical Factors Affecting the Efficiency of Some Sorting Techniques

Olabiyisi S.O., Adetunji A.B.

Dept. of Computer science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria,
Email: tundeolabiyisi@hotmail.com

Oyeyinka F.I.

Centre for Information Technology and Management, Yaba College of Technology, Lagos, Nigeria
Email: fkoyeyinka@yahoo.co.uk, kemi.oyeyinka@yabatech.edu.ng

Abstract— Sorting allows information or data to be put into a meaningful order. As efficiency is a major concern of computing, data are sorted in order to gain the efficiency in retrieving or searching tasks. The factors affecting the efficiency of shell, Heap, Bubble, Quick and Merge sorting techniques in terms of running time, memory usage and the number of exchanges were investigated. Experiment was conducted for the decision variables generated from algorithms implemented in Java programming and factor analysis by principal components of the obtained experimental data was carried out in order to estimate the contribution of each factor to the success of the sorting algorithms. Further statistical analysis was carried out to generate eigenvalue of the extracted factor and hence, a system of linear equations which was used to estimate the assessment of each factor of the sorting techniques was proposed. The study revealed that the main factor affecting these sorting techniques was time taken to sort. It contributed 97.842%, 97.693%, 89.351%, 98.336% and 90.480% for Bubble sort, Heap sort, Merge sort, Quick sort and Shell sort respectively. The number of swap came second contributing 1.587% for Bubble sort, 2.305% for Heap sort, 10.63% for Merge sort, 1.643% for Quick sort and 9.514% for Shell sort. The memory used was the least of the factors contributing negligible percentage for the five sorting techniques. It contributed 0.571% for Bubble sort, 0.002% for Heap sort, 0.011% for Merge sort, 0.021% for Quick sort and 0.006% for Shell sort.

Index Terms — Factor Analysis, Sorting techniques, Decision Variables, Eigenvalue, Principal Components, Community, Correlation

I. INTRODUCTION

In computer science study, much importance is attached to the way in which data are arranged. Data can be arranged in clustered form or grain (scattered or loosely) form. Information about data arrangement is very important because it is a predominant factor in accessing the data. Any data that is not easily accessible

to the user is not good and may be difficult to use. In other to make data useful and accessible at any given time the arrangement of the data in memory should be orderly. This orderly arrangement of data is referred to as sorting. Sorting can also mean the process of arranging items in sequence form and/or in different sets, and accordingly. Data can be sorted according to their kind, class, nature and so on. This ordering makes it possible or easy to search for a specific data or elements in the sorted list. Another important aspect in computer science is efficiency; therefore data are sorted in other to gain efficiency in retrieving or searching task.

Statistical analysis has been a useful tool in nearly all the fields of study, for example, Engineering, Science, Journalism, and Marketing. The field of statistics has been very useful when tasks involve estimating or analyzing values. The most recognized statistical package in use by many researchers is the Statistical Packages for Social Sciences (SPSS). There are several methods of data analysis in SPSS which include general linear model, generalized linear models, mixed models, compare means, dimension reduction, log-linear, non-parametric test, scales, survivals, forecasting, missing value analysis, multiple imputation, complex samples, quality control and so on.

In this work, factor analysis which is a branch of dimension reduction method is used. It seeks to discover if the observed variables can be interpreted in a more compressed form with few numbers of variables called factors. Sorting is an important process that determines the efficiency of many computing tasks and procedures. The speed of a particular sorting technique used in a task will determine how fast such task can be completed. Hence the efficiency of sorting technique affects to a large extent the efficiency of computing procedures [1].

This study evaluated the critical factors that affect the efficiency of sorting techniques. The objectives of the study are:

- (i) To carry out an exploratory study of critical factors affecting the efficiency of Shell, Bubble, Heap, Quick and Merge sorts.

- (ii) To conduct experiments in order to determine the efficiency of these sorting techniques mentioned in (i) above in terms of execution time, memory used and the number of exchanges/comparisons.
- (iii) To subject the result obtained in (ii) above to factor analysis by SPSS.

Factor analysis was first introduced by Thurstone in 1931. The main general purposes of introducing the techniques are to reduce the number of variables and to detect structure in the relationships between variables, i.e. to classify variables. Therefore, factor analysis is applied as data reduction or structure detection method. Factor analysis originated in psychometrics, and is used in behavioural sciences, social sciences, marketing, product management, operations research, and other applied sciences that deal with large quantities of data.

Factor analysis attempts to identify underlying variables or factors that explain the pattern of correlations within a set of observed variables. Factor analysis is often used in data reduction to identify a small number of factors that explain most of the variance observed in a much larger number or manifest variables. It is assumed that data should have a bivariate normal distribution for each pair of variables and observations should be independent. Using Factor analysis method under SPSS for specifying and capturing of the variables involves the following four procedural steps: Descriptive, Extraction, Rotation, Factor Scores, and Options methods [2]. Each of these categories also contains other alternatives to be selected depending on the type of analysis.

There are basically two types of factor analysis: Exploratory factor analysis (EFA) and Confirmatory factor analysis (CFA). Both types of factor analyses are based on the *Common Factor Model*. Exploratory Factor Analysis (EFA) attempts to discover the nature of the constructs influencing a set of responses. It is used to uncover the underlying structure of a relatively large set of variables. It is the common form of factor analysis. Confirmatory Factor Analysis (CFA) tests whether a specified set of constructs is influencing responses in a predicted way. It determines if the number of factors and the loadings of measured variables on them conform to what is expected on the basis of pre-established theory.

There are different types of factoring, these include principal factor also referred to as Principal Component Analysis (PCA). It seeks a linear combination of variables such that the maximum variance is extracted from the variables. It then removes this variance and seeks a second linear combination which explains the maximum proportion of the remaining variance, and so on. This is called the principal axis method and results in orthogonal (uncorrelated) factors. Another type of factoring is referred to as Canonical Factor Analysis (CFA): It is also called Rao's canonical factoring. CFA seeks factors which have the highest canonical correlation with the observed variables. CFA is unaffected by arbitrary rescaling in the data.

Other types of factoring include Principal Factor Analysis (PFA) or Principal Axis Factoring (PAF), Image Factoring, Alpha Factoring etc. The remaining part of this paper is organized as follows: Section II presents some literature review on sorting techniques and factor analysis, Section III describes the methodology, Section IV presents the result and Section V concludes the paper.

II. RELATED WORK

Reference [3] presented a paper on an exploratory study of factors affecting the efficiency of sorting techniques using shell, heap and treap in other to study their efficiency in terms of the running time and memory usage. The experimental results for the variables were generated from an algorithm implemented in JAVA in which the amounts of number sorted were varied for the different sorting techniques. Factor analysis by principal components of the obtained experimental data was carried out using Statistical package for Social Sciences (SPSS) for the purpose of estimating the contribution of each factor to the success of the sorting algorithms and one factor was extracted. Further statistical analysis was carried out to generate eigenvalue of the extracted factor. The eigenvalue formed the basis for estimating the contribution of the extracted factor. A system of linear equation was used to estimate the assessment of each assessor of the sorting techniques.

Reference [4] presented a paper on the overview of Factor Analysis. The paper stated the difference between EFA, PCA and CFA.

Reference [5] in their paper investigated sorting problems and their solutions. The paper explained the most popular algorithms that are useful for sorting lists. They are bubble sort, selection sort, insertion sort, shell sort, merge sort, heap sort, quick sort and bucket sort. Algorithms were represented with perfect descriptions and also tried to indicate their computational complexity in the worst, middle and best cases. At the end, implementation code was placed.

Reference [6] in their paper "Factor Analysis of the Performance Indices and Communications Technology Projects in Public Sector of the Nigerian Economy" proposed fifty three performance indices of ICT projects and a tool for evaluating them. The ICT projects in the public sector of Nigerian economy were surveyed and completed questionnaires were received from the entire sector. The data collected were subjected to factor analysis by principal components using SPSS.

Reference [7] in their paper "Development and Factor Analysis of Questionnaire to measure Patient Satisfaction with Injected and Inhaled Insulin for Type 1 Diabetes" used exploratory factor analysis to evaluate the data collated from the questionnaire.

The paper, Factor Analysis of the Effects of Academic Staff Profile on the Investment Portfolio of a University by [8] studied three factors; teaching,

research and community development as major investment foci of a university loaded on a number of academic profile related decision variables. The result obtained from the factor analysis provided a basis for deriving system of equations desirable for evaluating the contributions of each academic staff to teaching, research and community development.

The paper “An Empirical Model for Information Retrieval System Evaluation” by [9] took an in depth study of one of the major issue hindering easy access to information retrieval in other to come up with a better and efficient paradigm with much ease using a mathematical model and factor analytical method.

Reference [10] carried out factor analysis of post-implementation review of student information systems in Nigeria. In their paper, research was carried out in four Universities in other to evaluate the critical factors that may enhance the successes and failures encountered after implementation of the student information systems. An on-line questionnaire was designed using Macromedia Dreamweaver, PHP and Mysql to gather data from the users of the system. Analysis of the data gathered was carried out using SPSS 10 and Factor analysis was used for the assessment of system’s attributes identified.

III. MATERIALS AND METHODS

In the approach presented here, the efficiency of some sorting techniques was investigated using factor analysis by principal components. The major factors used in evaluating the sorting technique are time taken, number of swaps and total memory used. The sorting techniques investigated are Merge, Bubble, Quick, Shell and Heap. Experimental results for decision variables were generated by implementing the sort algorithms in Java.

Following [4], the decision variables of the impact of time taken, number of swap and memory usage relate to one another. The general form of the mathematical model for evaluating the decision variables is presented as:

$$Y_i = \sum_{k=1}^n a_{i,k} x_k, \quad i = 1, 2, 3, \dots, m \quad (1)$$

Where Y_i represents the i^{th} assessor’s observation of k^{th} decision variable, $a_{i,k}$ represents the assessment of k^{th} decision variable by i^{th} assessor. From equation 1 the mathematical model can be expressed by the system of equations below;

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + \dots + a_{1,5}x_5 \\ \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + a_{m,3}x_3 + \dots + a_{m,5}x_5 \end{pmatrix} \quad (2)$$

Where $x_k, k=1, 2, 3, \dots, n$ represents k^{th} decision variable, $a_{i,k}$ represents the assessment of the i^{th} sorting technique for the k^{th} decision variable .

The factor analysis by principal components was adopted in the evaluation of the decision variables of the impact of time, number of swap and memory consumed. The primary goal is to obtain the contribution of each of the factors to the efficiency of the sorting techniques. The following statistics were generated and used for this purpose: Descriptive statistics, Correlation matrix, Barlett’s test and Kaiser-Mayer Olkin (KMO), Communalities, Initial factor loadings, Rotated factor loadings, Factor score coefficient matrix, Eigen values.

In factor analysis there is a set of factors which is generally referred to as ‘common factors’ each of which loads on some of the values of the efficiency and another set of factors which are extraneous to each of the value of efficiency. The proportion of the variance of the values of efficiency explained by the common factor is called the ‘communality’ of the value of efficiency. The communality of the value of efficiency range between 0 and 1, where 0 indicates that the common factors explains none of the variance and 1 indicates that all the variance is explained by the common factors. The eigenvalues and the computed variance of each factor are used to determine the contribution of such factor to the efficiency of the sorting techniques. Statistically, a factor is efficient if the percentage variance contribution is 80% to 90% or more, anything lesser than this is rejected. However, some analysis allow as low as 50% to 70% depending on what the result is meant for. But in the context of this paper, the minimum percentage of fitness expected for the eigenvalues is 90%. The eigenvalue is used to indicate how well each of the factors fits the experimental data. The eigenvalue of j^{th} factor denoted by ‘ E_j ’ is calculated by:

$$E_j = \sum_{k=1}^3 X_{i,j}^2 \quad i=1,2,3; \quad j = 1 \quad (3)$$

Where $X_{i,j}$ represents the loading of j^{th} factor on i^{th} decision variable.

In an attempt to evaluate the percentage contribution of each factor to the efficiency of the sorting techniques, the eigenvalue of each factor is generated. The percentage

$$P = 100 \left(\frac{E_j}{n} \right) \quad (4)$$

Where n represents the number of decision variables considered in the study. Tables 5a to 5e present the eigenvalues, the percentage contribution and cumulative percentage contribution of the three factors considered for each of the sorting techniques.

IV. DISCUSSION OF RESULTS

A random number generator was incorporated into the Java program written for each of the five sorting techniques; Bubble sort, Quick sort, Shell sort, Merge sort and Heap sort. The numbers generated were varied from 2000, 4000, 6000, 8000 and 1000 for each of the sorting techniques. The program automatically prompted for how many random numbers to be generated, once it is generated the program automatically sort the data for all the five techniques.

The Java program also compiled the execution time in (nanosecond), the number of swaps and the memory used, in bytes, for each of the techniques. For instance, for 2000 random numbers generated;

Heap sort results are as follows:

- (i) the time taken in nanoseconds is 57295212,
- (ii) used memory (bytes) is 347352
- (iii) the number of swaps is 20095.

Merge sort results are:

- (i) the time taken in nanoseconds is 48080541,
- (ii) used memory (bytes) is 356248
- (iii) the number of swaps is 9574.

Quick sort results are:

- (i) the time taken in nanoseconds is 119460876,
- (ii) used memory (bytes) is 356784
- (iii) the number of swaps is 7050.

Bubble sort results are:

- (i) the time taken in nanoseconds is 111716893
- (ii) used memory (bytes) is 348248
- (iii) the number of swaps is 981057

Shell sort results are:

- (i) the time taken in nanoseconds is 85128026
- (ii) used memory (bytes) is 348704
- (iii) the number of swaps is 10289

A. Data Analysis Using Reduction Method

In statistics, the amount of data analyzed determine the authenticity of the result, this will make the analysis to become easier. Using SPSS to analyze data, it is expected that the number of data should be greater than or equal to 50 in order to obtain accurate results. Smaller data can be analyzed manually. In this work, three factors were analyzed. The three factors gave a total of seventy five data. The model of data analysis used in this work is as shown in figure 1, the problem was defined, principal component was used for factor model, and principal component analysis was adopted. The number of critical factor was determined using eigenvalues and scree test.

Factor analysis takes place in the factor analysis sub menu. Two important variables are selected, the Dependent variables and the Independent variables. Factors that can be used in estimating another variable is

dependent. In this work, time taken, number of swap and memory used are dependent variables while sort type is an independent variable.

The next stage is extraction; extraction methods include principal components, least square, maximum likelihood and factoring methods. In this work, principal components extraction method is used.

The analysis is done using correlation matrix and the display by unrotated factor solution and 'scree plot'. The extraction of the initial factor is based on eigenvalues which must be greater than 1.

The correlation matrix computes the correlation coefficients of the columns of a matrix. That is, row i and column j of the correlation matrix is the correlation between column i and column j of the original matrix. The diagonal elements of the correlation matrix will be 1 since they are the correlation of a column with itself. The correlation matrix is also symmetric since the correlation of column i with column j is the same as the correlation of column j with column i .

The analyzed result showed that each variable correlated perfectly with itself since the diagonal elements in each of them is 1.00 that is $r = 1$. There is high correlation among the variables for all the sorting techniques. Table I contains a summary of the correlations among the variables. For example, for bubble sort, the correlation between time taken and number of swap is 0.972, time taken and memory used is 0.953 and that of number of swap and memory utilization is 0.978 etc. Correlation should not be less than 0.5. The three factors showed high correlation in term of their loading on the five sorting techniques. Number of swaps and memory utilization correlated more for Bubble sort and Quick sort. The time taken has 100% correlation for all the five sorting techniques.

The component matrix presents the initial factor loadings [7]. The component matrix tables (table IIA to IIE) contain component loadings, which are the correlations between the variables and the components. The values range from -1 to +1 because they are correlations.

The columns under the component heading are the principal components that have been extracted. Three components were extracted. The components should not be interpreted the way that factors that have been extracted from a factor analysis are interpreted. Rather, the component scores are more useful because they are used for data reduction.

A study of Assortment of Different Sorting Algorithms was carried out by [11]. In this paper the effectiveness of some sorting techniques was compare in order to make computational becomes easy which will ultimately improves the throughput. The classifications, comparisons and differences between these sort techniques were compared.

This is the correlation between the factors and the components. The components with value less than 0.4 is ignored in this analysis hence components 2 and 3 were ignored.

Initial communalities are estimates of the variance in each variable accounted for by all components or factors. Extraction communalities are estimates of the variance in each variable accounted for by the factors (or components) in the factor solution. Small values indicate variables that do not fit well with the factor solution, and should possibly be dropped from the analysis.

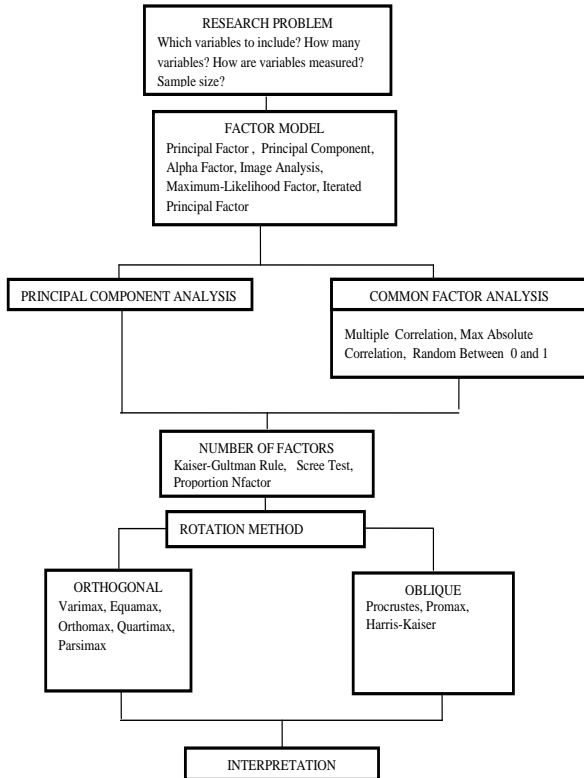


Figure 1: Factor Analysis Decision Diagram (adapted from[12])

TABLE I SUMMARY OF CORRELATION OF THE FACTORS AND THE SORTING TECHNIQUES

Factors	Time				
	Bubble Sort	Quick Sort	Heap Sort	Shell Sort	Merge Sort
Number of Swaps	0.972	0.961	0.952	0.802	0.757
Memory Utilization	0.953	0.965	0.944	0.765	0.755

TABLE IIA COMPONENT MATRIX FOR HEAP SORT

	Component		
	1	2	3
TIME TAKEN (NANOSECOND)	.976	.216	.000
NUMBER OF SWAPS	.996	-.094	-.006
MEMORY UTILIZATION	.993	-.118	.005

TABLE IIB COMPONENT MATRIX FOR BUBBLE SORT

	Component		
	1	2	3
TIME TAKEN (NANOSECOND)	.986	.163	.043
NUMBER OF SWAPS	.994	-.019	-.106
MEMORY UTILIZATION	.976	-.216	.013

TABLE IIC COMPONENT MATRIX FOR MERGE SORT

	Component		
	1	2	3
TIME TAKEN (NANOSECOND)	.879	.477	6.422E-5
NUMBER OF SWAPS	.977	-.213	-.013
MEMORY UTILIZATION	.988	-.143	.064

The communalities for the i th variable are computed by taking the sum of the squared loadings for that variable. This is expressed as:

$$C_i = \sum_{j=1}^m l_{ij}^2 \quad i=1, 2, 3, \dots, n \quad (5)$$

TABLE IID COMPONENT MATRIX FOR QUICK SORT

	Component		
	1	2	3
TIME TAKEN (NANOSECOND)	.983	.182	.001
NUMBER OF SWAPS	.995	-.096	.017
MEMORY UTILIZATION	.996	-.083	-.018

TABLE IIE COMPONENT MATRIX FOR SHELL SORT

	Component		
	1	2	3
TIME TAKEN (NANOSECOND)	.894	.447	.001
NUMBER OF SWAPS	.985	-.175	-.010
MEMORY UTILIZATION	.972	-.234	.009

TABLE III SUMMARY OF COMMUNALITY

Measures	Merge Sort	Bubble Sort	Heap Sort	Quick Sort	Shell Sort
Time Taken (Nanosecond)	1.000	1.000	1.000	1.000	1.000
Number of Swaps	1.000	1.000	1.000	1.000	1.000
Memory Utilization	1.000	1.000	1.000	1.000	1.000

One assessment of how well this model is doing can be obtained from the communalities. What should be looked for are values that are close to 1. This would indicate that the model explains most of the variations for those variables. From Table III, it is shown that all the values are close to 1 which indicates that the model explains the variations of the factors.

Any variable with communality less than 0.5 is remove from the analysis. Communalities are estimated values accounted for by the components. The values must be 1 or close to 1.

B. Determining the Number of Factors

Determining the optimal number of factors to extract is not a straightforward task since the decision is ultimately subjective. There are several criteria for the number of factors to be extracted, but these are just empirical guidelines rather than an exact quantitative solution. In practice, most factor analysts seldom use a single criterion to decide on the number of factors to extract. Some of the most commonly used guidelines are the Kaiser-Guttman rule, percentage of variance, the scree test, size of the residuals, and interpretability.

Percentage of Variance is related to the latent root criterion, it is the percentage or proportion of the common variance (defined by the sum of communality estimates) that is explained by successive factors. For example, if you set the cutting line at 75 percent of the common variance (proportion=.75 or percent=75), then factors will be extracted until the sum of eigenvalues for the retained factors exceeds 75 percent of the common variance, defined as the sum of initial communality estimates.

From Tables IVA to IVE, the percentage variance explained the contribution of each of the factors to the efficiency of the sorting techniques. The three factors contributed a total of 100% to the efficiency of the five sorting techniques, ‘time taken’ contributed 97.842%, ‘number of swap’ contributed 1.587% and ‘memory consumed’ contributed 0.571% impact on the efficiency of Bubble Sort. While for Heap sort, the ‘time taken’ contributed 97.693%, ‘number of swap’ contributed 2.305% and ‘memory consumed’ contributed 0.002%.

Also for Merge sort, ‘time taken’ contributed 89.351%, ‘number of swap’ contributed 10.638% and ‘memory consumed’ contributed 0.011%. For Quick sort, ‘time taken’ contributed 98.336%, ‘number of swap’ contributed 1.643% and ‘memory consumed’ contributed 0.021%. Finally, for Shell sort ‘time taken’ contributed 90.480%, ‘number of swap’ contributed 9.514% and ‘memory consumed’ contributed 0.006%. Table V shows the summary of the contributions of each factor.

In all the five sorting techniques considered, time taken and number of swap are critical. These results showed that time is the most critical factor affecting sorting techniques followed by the number of swap while memory used is negligible.

C. Scree Test

Sometimes plotting the eigenvalues against the corresponding factor numbers gives insight into the maximum number of factors to extract. Table VI contains the summary of the eigenvalues and is shown graphically in figure 2. The rate of decline tends to be fast for the first few factors but then levels off. The "elbow", or the point at which the curve bends, is considered to indicate the maximum number of factors to extract.

According to the scree plot in Figure 2, the point at which the curve begins to bend from the component are factor 1 and 2 and it level off at factor 3. The figure also illustrated that the eigenvalues drop rapidly at third factor, the decline in the eigenvalues gradually levels off. The scree plot suggests a maximum of two factors that are very critical in this study i.e. the time taken and the number of swap.

TABLE IVA TOTAL VARIANCE EXPLAINED FOR BUBBLE SORT

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings ^a
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	
	1	2.935	97.842	97.842	2.935	97.842	
2	.048	1.587	99.429	.048	1.587	99.429	2.677
3	.017	.571	100.000	.017	.571	100.000	2.437

TABLE IVB TOTAL VARIANCE EXPLAINED FOR HEAP SORT

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings ^a
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	
	1	2.931	97.693	97.693	2.931	97.693	
2	.069	2.305	99.998	.069	2.305	99.998	2.630
3	5.839E-5	.002	100.000	5.839E-5	.002	100.000	2.572

TABLE IVC TOTAL VARIANCE EXPLAINED FOR MERGE SORT

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings ^a
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	
	1	2.681	89.351	89.351	2.681	89.351	
2	.319	10.638	99.989	.319	10.638	99.989	2.085
3	.000	.011	100.000	.000	.011	100.000	.107

TABLE IVD TOTAL VARIANCE EXPLAINED FOR QUICK SORT

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings ^a
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	
	1	2.950	98.336	98.336	2.950	98.336	
2	.049	1.643	99.979	.049	1.643	99.979	2.676
3	.001	.001	100.000	.001	.001	100.000	100.000

TABLE IVE TOTAL VARIANCE EXPLAINED FOR SHELL SORT

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings ^a
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	
	1	2.714	90.480	90.480	2.714	90.480	
2	.285	9.514	99.994	.285	9.514	99.994	2.160
3	.000	.006	100.000	.000	.006	100.000	2.531

V. CONCLUSION

The efficiency with which sorting is carried out often has a significant impact on the overall efficiency of a program. The factors that affect the efficiency of Bubble, Heap, Merge, Quick and Shell sorting techniques in terms of running time, number of swaps and memory used was studied. Experiments were conducted and results were subjected to factor analysis by SPSS. The time taken to sort, number of swaps and memory used was used as decision variables to evaluate their efficiencies. Experimental results for the decision variables were generated from software tools in which the amount of number sorted were varied for the five sorting techniques. The results were subjected to factor analysis using SPSS to test the level at which each of the factors affect the sorting techniques. Eigenvalues were used to indicate how well each of the decision variables

fits the data from the experimental results. From the results obtained, the main factor affecting the sorting techniques was time taken to sort. It contributed 97.842%, 97.693%, 89.351%, 98.336% and 90.480% for Bubble sort, Heap sort, Merge sort, Quick sort and Shell sort respectively. The number of swap came second contributing 1.587% for Bubble sort, 2.305% for Heap sort, 10.63% for Merge sort, 1.643% for Quick sort and 9.514% for Shell sort. The memory usage was the least of the factors contributing negligible percentage for the five sorting techniques. It contributed 0.571% for Bubble sort, 0.002% for Heap sort, 0.011% for Merge sort, 0.021% for Quick sort and 0.006% for Shell sort.

Time taken to sort is the main factor affecting the efficiency of sorting techniques. It was observed from the experiment carried out that merge sort has the least run time among the five sorting techniques for smaller dataset. When the number of data grew beyond 5000, Merge sort had the least sort time. Quick sort had the worst run time out of all sort techniques considered. For number of swaps, Bubble sort was the most efficient as it has the least number of exchanges for all data set and Heap sort was observed to be most efficient in memory utilization. In conclusion, Merge sort is the most efficient sorting technique when considering the most critical factor affecting sorting techniques. However, eigenvalues and the scree plot strengthen the fact that the running time and the number of swaps are the most critical factors affecting the efficiency of the sorting techniques and the most efficient sorting techniques based on these findings are Merge sort and Bubble sort respectively.

The result of this research work reveals that the factors of time and number of swaps are critical to the efficiency of sorting techniques. It is recommended that people that build or work with sorting techniques should consider the results obtained in this work when choosing or using sorting techniques. In time critical applications, the sort time of different sorting techniques may contribute largely to the success of such application. However, other factors that have high correlation with run time may also affect the application efficiency.

In future work, system environment and software factors could be explored as other factors affecting sorting by these methods. There are other factor analysis methods that could also be used for data analysis of this kind. The method is called General linear model, it is suitable when analyzing experimental data. On the other hand, Reduction method is suitable for analyzing data collected using questionnaire. It is therefore suggested that further research study should be carried out in this area using general linear model. This paper investigated five sorting techniques and three factors. The number of sorting methods could be increased to six, seven, eight or more to get better results.

TABLE V SUMMARY OF EIGENVALUES

Components	Quick Sort	Bubble Sort	Heap Sort	Shell Sort	Merge Sort
1	98.336	97.842	97.693	90.480	89.531
2	1.643	1.587	2.305	9.514	10.638
3	0.021	0.571	0.002	0.006	0.011

TABLE VI COMPARISON OF SORTING TECHNIQUES TOTAL VALUES

Technique	Components		
	1	2	3
Bubble Sort	2.935	0.048	0.017
Heap Sort	2.931	0.069	5.84E-05
Merge Sort	2.6810	0.3190	0.0000
Quick Sort	2.95	0.049	0.001
Shell Sort	2.7140	0.2850	0.0000

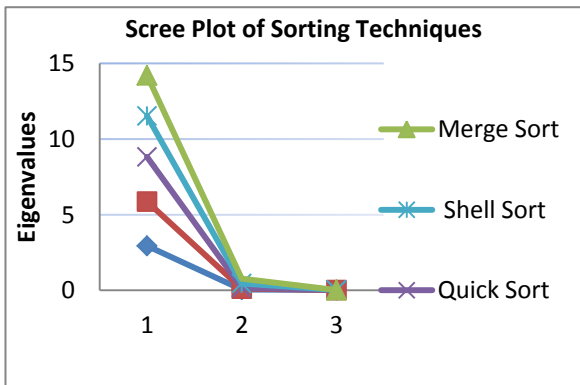


Figure 2: Scree Plot of the Eigenvalues of the Sorting Techniques

REFERENCES

[1] Chen C., Khoo P. And Yan W.,(2006) – An investigation into effective design using sorting technique and Kohonenself organising map, *Advances in Engineering Software*, Vol.37, Issue 5, 334-349

[2] Hatcher L., (1994).*A step-by-step approach to using the SAS system for factor analysis and structural equation modeling*. Cary, NC: SAS Institute Press.

[3] Folorunsho O., Vicent O., Salako O.,(2010) - An Exploratory Study of Critical Factors Affecting the Efficiency of Sorting Techniques (Shell, Heap and Treap). *Anale.SeriaInformatica*. Vol. VIII fasc. 1.

[4] DeCoster, J., (1998).Overview of Factor Analysis.Retrieved (August 10, 2011) from <http://www.stat-help.com/notes.html>.

[5] Canaan C., Garai M.,andDaya M.,(2011) – Popular Sorting Algorithms, *Information instituteChireddzi, Zimbabwe World Applied Programming*, Vol (1), No (1),42-50, ISSN: 2222-2510.

[6] Akinyokun O., Angaye O., and Ubaru O. (2009) - Factor Analysis of the Performance Indices and Communications Technology Projects in Public Sector of the Nigerian Economy. *Journal of Technology Research*, Vol. 1.

[7] Cappelleri C., Gerber A., Kourides A., and Gelfand A., (2000) – Development and Factor Analysis of Questionnaire to measure Patient Satisfaction with Injected and Inhaled Insulin for Type 1 Diabetes. Pfizer Inc. (J.C.C., R.A. Ger., R.A. Gel.), Global Research and Development, Groton, Connecticut; and Pfizer Inc. (I.A.K.), New York.

[8] Akinyokun O., Uzoka F.(2007) –Factor Analysis of the Effects of Academic Staff Profile on the Investment Portfolio of a University. *International Journal of The Computer, the Internet and Management* Vol 15#1.

[9] Akhigbe B., Afolabi B., and Adagunodo E.(2011) – An Emperical Model for Information Retireval System Evaluation: The User’s perspective. *Computer Engineering and Intelligent System* Vol 2 No. 4.

[10] Chiemeké C., and Osazuwa W.,(2008) – Factor Analysis of Post-Implementation Review of Student Information Systems in Nigeria. *Research Journal for Applied Science* Vol. 3 No 5.

[11] Vandana., Singh A., Monika.,&Kaur S., (2011) – “Assortment of Different Sorting Algorithms”.*Asian Journal of Computer science and Technology*, Swami Vivekanand Engineering College, Punjab, India.

[12] Introduction to SAS. UCLA (1995): Academic Technology Services, Statistical Consulting Group. from <http://www.ats.ucla.edu/stat/sas/notes2/> (accessed May 10, 2012).



Dr. Olabiyisi S.O. holds a Ph.D. in Computer Science and he is a Senior Lecturer at the Department of Computer Sciences and Engineering, Ladoke Akintola University of Technology, Ogbomosho, Nigeria.



Mrs. Oyeyinka F.I. is a Masters Student at the Department of Computer Sciences and Engineering, Ladoke Akintola University of Technology, Ogbomosho, Nigeria. She works as an assistant Chief Programmer at the Yaba College of Technology, Lagos, Nigeria.

Dr. (Mrs.) Adetunji A.B. holds a Ph.D. in Computer Science and she is a Lecturer at the Department of Computer Sciences and Engineering, Ladoke Akintola University of Technology, Ogbomosho, Nigeria.