



AN APPROACH TO SAVE ENERGY CONSUMPTION FOR SMARTPHONE APPLICATION DEVELOPMENT USING SUITABLE SORTING ALGORITHM

Murlidhar Verma

Research Scholar

Department of Computer Science & Engineering
MBM Engg. College, JNV University, Jodhpur, India

K.R. Chowdhary

Professor & Head (Retd.)

Department of Computer Science & Engineering
MBM Engg. College, JNV University, Jodhpur, India

Abstract: Smartphone has become the most important part of everyone's life. Smartphone has influenced every part of the life. It has enhanced and advanced professional and personal life. It has become a vital part of the life. There is no limit what one can do with a Smartphone these days. There is an App for almost anything. One can do online ticket booking, mobile banking, checking of the nearest coffee shop, reading the online newspaper, e-books, finding a path on maps etc. Today's Smartphones have the higher configuration in terms of CPU power, memory capacity, and storage but their functionality is limited by low battery power capacity.

The battery power of a Smartphone may be prolonged by optimizing hardware or software. This paper presents an approach for energy saving through software by choosing the appropriate sorting algorithm. Energy consumption of four sorting algorithms: Bubble sort, Insertion sort, Quick sort and Selection sort were measured. The results show that Quick sort is the energy efficient sorting method on average cases. Bubble sort is the most energy consuming algorithm. This work provides a general guideline to select appropriate sorting algorithm in development of energy efficient Smartphone application.

Keywords: smartphone; sorting; energy consumption;

I. INTRODUCTION

Smartphone has become the most important part of everyone's life. Smartphone has influenced every part of the life. It has enhanced and advanced professional and personal life. It has become a vital part of the life. There is no limit what one can do with a Smartphone these days. There is an App for almost anything [1]. One can do online ticket booking, mobile banking, checking of the nearest coffee shop, reading the online newspaper, e-books, finding a path on maps etc.

Today's Smartphones have a higher configuration in terms of CPU power, memory capacity, and storage but their functionality is limited by low battery power capacity. The battery power of a Smartphone may be prolonged by optimizing hardware or software. Many efforts have been made by Researchers to reduce energy consumption by optimizing hardware [2][3]. However, energy consumption may be optimized by using software-based approach i.e. by using an appropriate algorithm.

This paper presents an approach for energy saving through software by choosing the appropriate sorting algorithm. Here sorting algorithm is used because sorting algorithm efficiency is relevant to almost all applications. Sorting algorithms play a major role regarding performance and energy consumption of mobile devices.

Energy consumption of four sorting algorithms: Bubble sort, Insertion sort, Quick sort and Selection sort were measured. The results show that Quick sort is the energy efficient sorting method on average cases. Bubble sort is the most energy consuming algorithm. This work provides a general guideline to select appropriate sorting algorithm in development of energy efficient Smartphone application.

The remainder of this paper is structured as follows: Next section II gives an overview of sorting algorithms, the experiment and results are presented in section III, Section IV summarizes and concludes the paper.

II. SORTING ALGORITHMS

This section briefly describes the sorting algorithms and their time and space complexities. Almost all the applications use sorting algorithms. For the mobile system application development, there is a requirement of energy efficient sorting algorithms because these are battery-powered devices and battery power is limited. Apart from time and space complexities the energy requirement of any sorting algorithm is not properly known yet. This work tries to find out energy consumptions of popular sorting algorithms like Bubble, Selection, Insertion and Quick sort [5].

Bubble sort is easy and simple to implement. In this sorting method, each pair of adjacent elements is compared and swapped if they are in the wrong order. Each element is compared with its adjacent element, If the first element is larger than the second one then the position of the elements are interchanged, otherwise, it is not changed. The same process is repeated for all the elements of the List. In the one scan of the List, one element is placed at its correct position. This sorting technique is called bubble sort because items in the List move up into the correct order like bubbles rising to the surface. For the list of n elements $n-1$ passes are required and in each pass, there is n -pass number of comparisons needed to sort the List. Worst-case and average case complexity is $O(n^2)$.

```

Algorithm: BubbleSort (A, N)
Repeat for I=1 to N
  Repeat for J=1 to N-I
  If(A[J]> A[J+1])
  Then: Swap(A[J] and A[J+1])
return

```

Insertion sort works like playing bridge game of cards. In this sorting technique, a list of elements is sorted by inserting an element into an existing sorted list [5]. For example, There is one array A with N values A[1], A[2],...A[N]. The Insertion sort algorithm scans it from A[1] to A[N], inserting each element A[J] into its proper position in previously sorted subarray A[1], A[2], A[J-1]. It has the average time complexity of $O(n^2)$. It is also known as In-place sorting algorithm because it requires a constant amount of extra memory space $O(1)$.

```

Algorithm: InsertionSort (A, N)
Repeat for I=1 to N
  Temp= A[I+1]
  Set J=I+1
  While J>0 and A[J-1]>Temp
    A[J]=A[J-1]
    J=J-1
  End while
  A[J]=Temp
return

```

Selection sort is an in-place comparison-based algorithm. This algorithm searches the minimum value in the list and exchanges it with the first element. The same process is repeated for the remaining values in the list to sort the final list. The time complexity of the selection sort is $O(n^2)$.

```

Algorithm: SelectionSort(A,N)
Repeat for I=1 to N
  Repeat for J=I+1 to N
  If ( A[I]>A[J])
  Swap(A[I],A[J])
Return

```

Quick sort is also known as partition-exchange sort. It is based on divide and conquer technique. In this, a large list of numbers is partitioned into two sub-lists one of which contains values smaller than the specified value, say pivot, based on which the partition is made and another sub-list holds values greater than the pivot value. Quick sort calls itself recursively to sort the list.

The average case time complexity of the Quick sort is $O(n \log n)$. In the worst case, its time complexity is $O(n^2)$. The worst case occurs when the list is already sorted.

```

Algorithm: QuickSort(A,LB,UB)
If(LB<UB)
  Set pivot=LB
  I=LB
  J=UB
Repeat While(I<J)
  Repeat while ( A[I]<=A[pivot] AND I<UB)
  Update I=I+1
  Repeat While ( A[J]>A[pivot])
  Update J=J-1
  If (I<J)
  Swap(A[I],A[J])
Swap(A[pivot],A[J])
Call QuickSort(A,LB,J-1)
Call QuickSort(A,J+1,UB)

```

III. EXPERIMENT

Energy consumption of four sorting algorithms: Bubble sort, Insertion sort, Quick sort and Selection sort were measured. To measure the energy consumption of each algorithm four different Apps were developed for each sorting algorithm.

Power Tutor V.1.4 App was used to measure the energy consumption of each sorting algorithm. Once Sorting App was started there was no human interaction with the Smartphone. To find accurate result each algorithm was run 10 times and average energy consumption was noted. Samsung GT-I8262 and Samsung J5 Prime Smartphones were used to run and measure the energy consumption of sorting algorithms. Almost same results were obtained on both the Smartphones. 10 different datasets of random numbers were used to find the results correctly.

Random numbers of different size as shown in TABLE-I were generated and the same numbers were input to the different sorting algorithms and energy consumption was measured. The graph in Fig.1. shows trends of experiments performed on Samsung J5 Prime.

From The Graph and The Table, it shows that Quick Sort is the most energy efficient algorithm whereas Bubble Sort consumes the higher amount of energy.

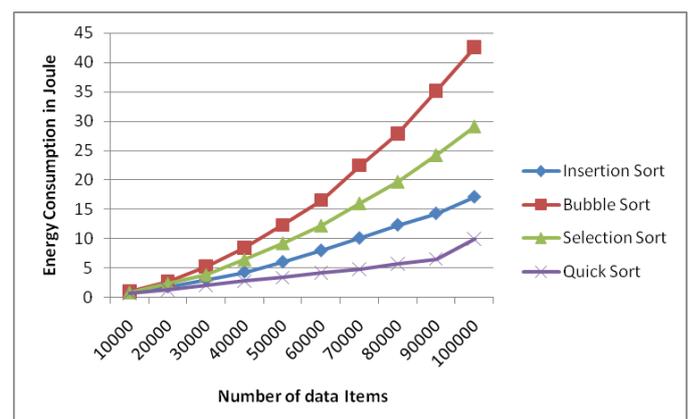


Fig.1: Energy Consumption of Sorting Algorithms

TABLE I. ENERGY CONSUMPTION IN JOULE

Data Size	Insertion Sort	Bubble Sort	Selection Sort	Quick Sort
10000	0.755	0.99	0.848	0.696
20000	1.78	2.7	2.37	1.37
30000	3.03	5.28	3.86	2.1
40000	4.3	8.43	6.42	2.75
50000	6.03	12.33	9.23	3.38
60000	7.98	16.54	12.2	4.2
70000	10.1	22.45	15.98	4.8
80000	12.35	27.87	19.66	5.67
90000	14.32	35.08	24.18	6.52
100000	17.14	42.57	29.04	9.96

IV. DISCUSSION

Even though there are many methods to measure energy consumption in Smartphones and mobile devices. In this work, App-based approach was used for simplicity. *Power Tutor V.1.4* App is used to find the energy consumption of each sorting App. This gives the relatively correct result because the same App is used to measure the energy consumption of all the sorting Apps and it was repeated 10 times. The Same randomly generated data set was input to all the sorting Apps.

Normally it is assumed that recursive calls in quick sort will consume more energy but, from results, it was observed that Quick sort is the most energy efficient and Bubble sort is the most energy consuming sorting algorithm. This might be due to multiple cores.

V. CONCLUSION AND SUMMARY

Developing energy efficient applications is really a challenging task. By using appropriate algorithm energy

consumption may be reduced. This work shows that Quick sort is the most energy efficient algorithm on average case. Bubble sort is the most energy consuming. Sorting algorithms were used because these are used in almost all the applications. This work can be further extended for other algorithms.

This work provides general guild line for selecting appropriate sorting algorithm in designing of energy efficient applications.

VI. ACKNOWLEDGMENT

I would like to express my deep sense of gratitude and wholehearted thanks to my guide Dr. K.R. Chowdhary Sir, for giving his valuable guidance, inspiration and encouragement to complete this paper. Without his support, guidance, and assistance this paper would not have been completed.

VII. REFERENCES

- [1] Murlidhar Verma, and K.R. Chowdhary, "Smartphone as a Tool for Different Applications", International Journal of Innovative Research in Science & Technology(IJRST), vol. 2, Issue 7, pp. 89–92, December 2015, ISSN (online): 2349-6010.
- [2] J. J. Chen and L. Thiele, "Expected system energy Consumption minimization in leakage-aware DVS Systems", In Proc. of the 13th ISLPED, pages 315-320 ACM, 2008.
- [3] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90nm low-power FPGA for battery-powered applications", In Proc. of the ACM / SIGDA 14th intern. symposium on field programmable gate arrays, pages 3-11 ACM,
- [4] Christian Bunse, Hagen Höpfner, Essam Mansour, Suman Roychoudhury, "Exploring the Energy Consumption of Sorting Algorithms in Embedded and Mobile Environment", Tenth International Conference on Mobile Data Management Systems, Services, and Middleware, pp. 600- 607, 2009.
- [5] Aaron M Tenenbaum, Yedidyah Langsan and Moshe J. Augenstein, "Data Structures Using C ", Pearson Education India.