



# IMPLEMENTATION OF ONLINE BIDDING SYSTEM WITH LIVE AUCTION USING IMPROVISED SORTING TECHNIQUE

Varnika Tyagi  
Department of IT  
Inderprastha Engineering College,  
Ghaziabad,  
Uttar Pradesh, India.

Taruna Chopra  
Department of IT  
Inderprastha Engineering College,  
Ghaziabad,  
Uttar Pradesh, India

Tanya Aneja  
Department of IT  
Inderprastha Engineering College,  
Ghaziabad,  
Uttar Pradesh, India.

Vidhi Jain  
Department of IT  
Inderprastha Engineering College,  
Ghaziabad,  
Uttar Pradesh, India.

Ms. Chhavi Sharma  
Department of IT  
Inderprastha Engineering College, Ghaziabad,  
Uttar Pradesh, India.

**Abstract— The Internet has derived globalization among the people, different business institutes and lots more. Online Bidding System is a prominent solution to meet the expectations of online buyers and sellers where the products are often obtained at a reasonable price by both seller and buyer. The Online Bidding System is implemented via Live Auction where several bidders can bid at a point of time. These bidders can bid together over any product. In order to extract the highest bid from all received bids so as to update the current bidding amount on which further bidding process stands, a sorting is required. Although there are lots of sorting techniques like Quick Sort, Merge Sort etc. are available but all these algorithms takes large amount of time to derive the highest bid recursively. In paper we tried to present an improvised solution of sorting algorithm in the bidding process serving the implementation of a Live Auction. The extraction of the sorting algorithm is done by keeping in reference to the existing algorithms. This extracted algorithm is known as partial quicksort whose experimental results shows that it sorts the large amount of data in less execution time compared to other conventional sorting algorithms and further showing the working of Live Auction.**

**Keywords— *Partial Quick Sorting, Live Auction, Redundancy Check, Time Stamp.***

## I. INTRODUCTION

Online auctions are a mirror of traditional auctions and typically include multiple bidder participation. Auction mostly meant the sale of products or property where people make higher and better bids for every product until that gets sold to the one who pays the foremost, which is referred to as an English auction. Palpable and intangible products are bought and sold by bidders and sellers in both scenarios. Starting bids are low but increase with due time to fulfill the demand of the market and popularity of item. The time period of an internet auction ranges from one to 10 days for items offered 24/7 across the world. In actuality, apart from English auction, there are many varieties of auctions like Dutch, Vickrey so many depending upon factors, sort of a process or price with which the auction initiates, the time it takes to finish up it, restrictions on bid amount so on. It's meant to maintain a proper check for the safety between the admin and the users. Online Auctions are a major acceptance to the business models for the subsequent reasons:

- No fixed time constraint
- Flexible points in time
- No geographical limitations
- Offers highly intensive social interactions
- Includes an oversized number of sellers and bidders, which inspires a high-volume online business.



The main contributions of this paper are given below:

- To propose the sorting technique this is better than the conventional sorting algorithms present.
- To boost the efficiency and effectiveness of the Live auction System by implementing it with the proposed sorting technique.

## II. PROPOSED ALGORITHM

### A. Literature survey-

Online Bidding system is the foremost important e-business application which impacts on every consumer and business field. Day by day the usage of online bidding systems is increasing. The time is becoming digital since consumers are also moving towards the e-sites for purchase and sale. To handle Bidding, prior requirement is to debate about auction/bidding system and its type. Also to debate about the issues occurring, then mainly concentrate on online auction as possible solutions to a reference [8].

In keeping the view of people's demand the Survey on Online Auction System describes the overview of current going auction forms and therefore the other related issues like designing of an effective system for predicting the end bid price and the problem faced by online auction system i.e. skill bidding.[5]

In spite of its popularity among the multiple users the selling, bidding and buying of products using online auction websites, the prevailing literature gives little understanding of what causes users to settle on one auction site over another. A technique to look at this is also used to explore users' satisfaction with auction websites. A wider view of understanding of bidder satisfaction can facilitate the model of a customer-friendly client interface, thereby increasing the efficiency and effectiveness of auction websites and further leading to the success of e-commerce applications and businesses. [9]

A Web-Based System for Online Auction has two parts- Seller and Buyer module collectively called as Customer Interface and Admin Interface. Customer Module allows a customer to upload a particular product available under a selected category and bid on a selected product to shop for. Construction of this technique presents a web display of division of products based on the category they require to sell or bid. There's an Admin Module where an admin controls the entire bidding system. Admin endorse products based on the categories and looks over the registered customers. There's a fixed delivery policy. After completion of the bidding process, there's a notification system to notify the sellers and bidders. [6]

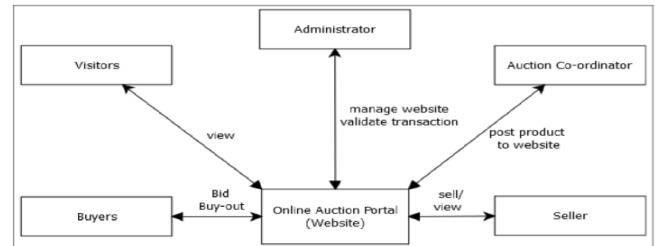


Fig. 1. System Architecture

In recent years, the important paragon in e-advertising is a real-time based bidding. It permits advertisers to purchase individual ad impressions through a real-time based auction in order to obtain maximum revenue. This strategy of bidding can be modified for the real time auction of the products instead of the advertisements. Thus, gives an idea of Live Auction implementation in Online Bidding System where several bidders around the globe can bid on the product simultaneously at a point of time.[1]

Partial Sorting is when we need to obtain a sorted list of the  $n$  smallest elements from  $n$  element set in some listed applications. This problem is known as partial sorting. The whole array can obviously be sorted all together at once but that amount of work is not necessary [10].

### B. Proposed System-

Several works exist regarding the solutions to the Online Bidding System. The work uses several modules in order to implement the auction. There is a login module which registers the user and seller to the portal and provides a membership to perform respective operations.

The portal provides a service of selling and buying goods online. The seller can put the goods in a sale for a default period of time that remains the same for every sale. The bidder will start bidding with the price set by the seller. The auction will run for a limited time with a sorting to sort the best price of the day. This sorted price will become the bidding price for the next bid. This sorted result leads to the winner of sale in the end. If there occurs a large number of bidders with the same bid amount then the seller has freedom either to select the winner of auction manually or use the Live auction feature to get buyers.

### Working

Once the product is registered for bidding then bid price becomes the only parameter considered for winning the auction. A bid price can't be any negative real number. The bid price represents the value, the bidder is wishing to pay for the product within the event. Once all bidders interested in the bidding (i.e., the bid set) have been identified a number of parameters may be used to select a winning bidder.



In order to determine the winning bidder we have build a small auction service. In this service, the auction should perform the underlying operations:

- Place a suitable bid
- Get the current highest bid
- Finish the auction process

In order to conduct the Live Auction the seller will begin with the initial parameters. The price set by the seller is modeled as an initial Bid on the registered product. The pool of bidders having wish to participate in the auction will show their presence by making themselves ACTIVE. When a new bid is registered it is yet to be decided whether the new bid is the winning bid or not. The state needs to be updated accordingly. If the registered new bid is larger than the currently setted bid then it will become the current Bid and a higher bid needs to be offered to win the auction. The Live Auction features the participation of multiple bidders with increasing values of bids with due time for a default time period. Hence, a continuous stream of bids for the registered product is received.

The bids would be placed in the concurrent manner at a point of time by these multiple bidders. These numerous bids are proctored using sorting technique which results to the highest bid as a current bid of auction. If there is a tie between the highest two bids then the bid that seemed to be registered earlier wins. The bid is registered to keep a check of the timestamp.

If there's a tie between the timestamp, the auction will terminate when default time comes to an end and the winner will be manually selected by the seller.

### Sorting Technique in Live Auction

A sorting rule is an algorithm that places elements of a list in a specified order. During the auction process, concurrent multiple bids are received. These bids are invigilated to get the highest bid. For such a process the sorting technique is opted. Multiple sorting algorithms exist but not limited to Bubble ort, Selection Sort, Merge Sort, and Quick sort. The sorting algorithms provided have certain advantages and disadvantages when being compared with other sorting algorithms. Say for an instance, classification of algorithms for sorting can be done according to their computational complexity, memory usage, stability, and adaptability, among other factors.

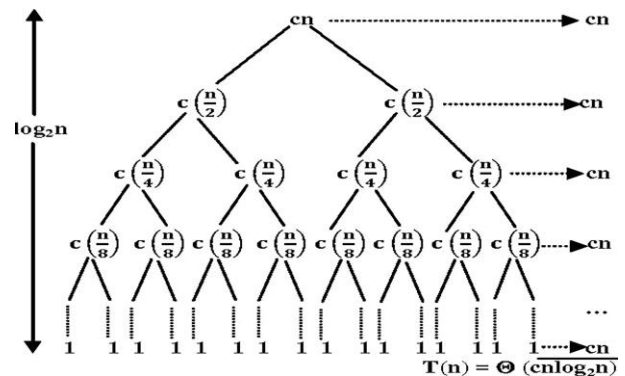
Some efficient algorithms that are commonly used for this task are namely Quick and Merge Sort as studied from M. N. Kumar et al.(2014). Both of these are recursive algorithms, which implements a three-step process to sort a subarray

A [p . . . r]:

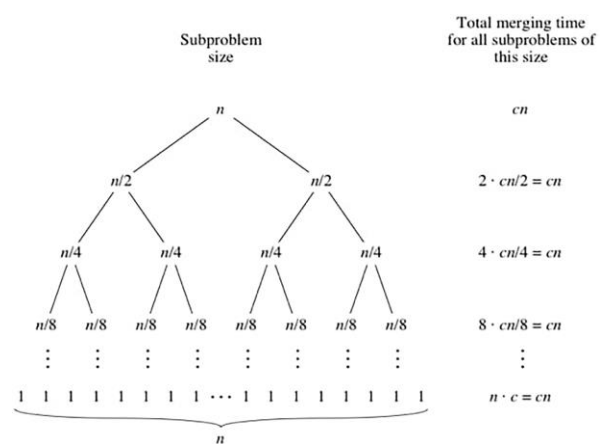
- (1) divide
- (2) conquer
- (3) combine

Although Quick sort may be useful in numerous applications due to its good cache performance for every cache level. Though the conventional Quick sort algorithm is in place but still introduce certain inefficiencies as in the pool of bidding data , sorting of each and every element in a data set may be unnecessary making it unusable for the large amount of data sets because quick sort yields  $O(n^2)$  as its worst case complexity as it requires a lot of comparisons.

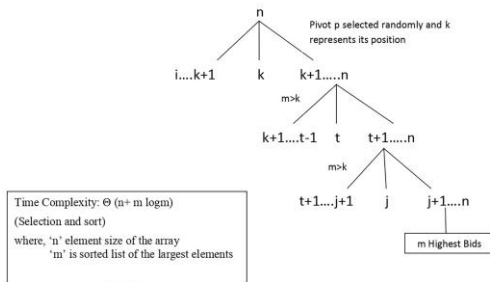
Whereas Merge sort being an external algorithm can be implemented efficiently on any type of data sets irrespective of its size (either large or small) having worst and average case complexities as  $O(n \log n)$ . Merge sort is not accurate because it requires additional memory space to store the auxiliary arrays because it uses three arrays amidst which two are used for retaining the given halves, while the third external array stores the final list of sorted elements as it merges the other two lists and then sorting of each array is done recursively. Lastly, the sub arrays are taken to merge in order to to visualize it as the 'n' element size of the produced array respectively.



(a) Quick Sort



(b) Merge Sort



(c) Partial Quicksort

Fig. 2. Recursion Trees (a) Quick Sort (b) Merge Sort (c) Partial QuickSort

Implementing the Merge sort and Quick Sort in this application would not be a suitable choice as only a small subset of higher priced bids need be considered as candidates for the bidding. Thus, eliminating the sorting of lower priced bids may provide for the determination of the winning bid more rapidly. For its up gradation, we refer to a simple variation of quick sort which helps in effective solving of the problem. In our application, we need to list of the  $m$  largest elements of a given set of  $n$  elements as shown in Fig.2.(c). This solution is known as *partial sorting*.

Sorting the entire array at once is undoubtedly a convenient solution but it provides the results and outputs which are not even desired. A solution, given by Conrado Martínez et al.(2004), 'partial quick sort', a simple, efficient and elegant variant of quick sort which is summarized by and Marianne Durand et al.(2005) that resolves the problem of partial sorting, by integrating selection and sorting operations together into a single algorithm. Using cost-efficient algorithms for both the tasks (selection as well as sorting) the total cost is  $T(n + m \log m)$ . This algorithm uses the principles of quicksort and proceeds as follows :

Through recursive call, the value ' $m$ ' and a sub array  $A[i..j]$  is received such that  $i = m$ . The sub array must be rearranged resulting to which  $A[m..n-1]$  contains the  $n - 1 - m$  largest elements belonging to  $A[i..j]$  in ascending order; if  $m > j$  which means that we must fully sort  $A[i..j]$ . The recursive call begins with  $A[1..n]$  such that if there are almost no elements then the completion of work is successful. Otherwise, one of the elements from the dataset is chosen as the pivot element, i.e.  $p$ , thereby partitioning the array  $A[i..j]$  into  $A[i..k-1]$  that comprises elements that are smaller than  $p$ ,  $A[k]$  contains the element chosen as pivot i.e.  $p$  and  $A[k+1..j]$  comprises of the set of elements which are greater than the pivot.

Suppose that there are exactly  $n-1$  comparisons of elements required to carry out the partitioning process in the array such that the partitioning process conserves randomness. In order to get the largest  $m$  elements, the  $m^{\text{th}}$  index will be required which can be derived by  $n-m$  and can be stored in any variable say ' $d$ '.

Now if  $k < d$  then recursive calls will occur in the right subarray. If  $k > d$  then recursive calls will occur in the left subarray in order to meet the termination condition. If  $d = k$  then quick sort will be applied to the right subarray without any further random calls which will result to list of  $m^{\text{th}}$  largest bids. [10].

```
void partial_quicksort (int[] A, int i, int j, int d)
{
    if (i < j)
    {
        int k = find_pivot(A, i, j); //generated randomly
        d = partition(A, d, i, j, k); //partitioned
        if (k < d)
        {
            partial_quicksort(A, k + 1, j, d);
        }
        if (k > d)
        {
            partial_quicksort(A, d, j, d);
        }
    }
}
```

**Algorithm of Partial Quick Sort**

**Comparison of Quick Sort, Merge Sort with Partial Quick Sort**

In this Section, we have created a comparison table of Time and Space complexity for conventional Quick Sort, Merge Sort and Partial Quick Sort. The studies are shown below:

Table-1 Performance Analysis

Complexities	Quick Sort	Merge Sort	Partial QuickSort (expected)
<b>Best Case Time Complexity</b>	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n + m \log_2 m)$
<b>Average Case Time Complexity</b>	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n + m \log_2 m)$
<b>Worst Case Time Complexity</b>	$O(n^2)$	$O(n \log_2 n)$	$O(n + m \log_2 m)$
<b>Space Complexity</b>	$O(\log_2 n)$	$O(n)$	$O(\log_2 n)$



The result driven from the Table 1 states that Partial QuickSort is efficient than the other two algorithms based on Time and Space Complexities such that the Time Complexity  $O(n+m \log_2 m) \sim O(n \log_2 n)$ . Also the Merge Sort has greater Space Complexity compared to Partial QuickSort.

Based on the above results, Runtimes of the algorithms for a specific number of Bids are computed. The Runtime for each bid is computed by taking the average of four consecutive value of execution time for all runs to get the accurate reading. The Result for all runs of each algorithm and number of Bids are shown in Table 2 and Fig.3.

Table- 2 Runtime Analysis of Quick Sort, Merge Sort and Partial QuickSort

Bids	Execution Time(average time in seconds)		
	Quick Sort	Merge Sort	Partial Quicksort
1000	0.15	0.16	0.15
5000	0.21	0.29	0.22
10,000	0.36	0.48	0.36
20,000	0.44	0.53	0.43
30,000	0.49	0.55	0.47
40,000	0.62	0.63	0.56
50,000	0.61	0.64	0.59

In the below Fig.3, On the x-axis we have placed the Number of Bids and on the y-axis we have taken the execution time in seconds. The Execution Time includes the printing of sorted result.

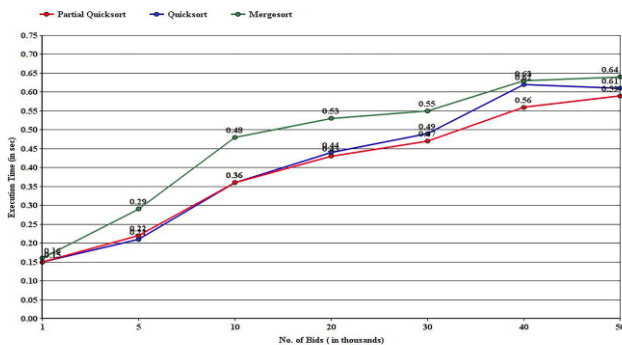


Fig. 3. This is a plot of Time Comparison of Quick Sort(-),Merge Sort(-) and Partial QuickSort (-) over the random data generated by random number generator. Through the Fig.3, it can be concluded that Partial QuickSort is faster than the other sorts.

**Comparison of Partial QuickSort with Randomized QuickSort**

In the Partial QuickSort, the pivot was generated randomly which was not generated randomly in conventional quick sort.

Therefore, we again compared the Partial QuickSort with the Randomized QuickSort. Here the pivot was generated randomly in both the Sorting Techniques. Beside the fact, Both the sorting techniques have same worst case complexities still the execution time of partial quicksort is always less than the execution time of Randomized QuickSort for each and every bid. The Runtime for each bid is computed by taking the average of four consecutive value of execution time for all runs to get the accurate reading. The Result for all runs of each algorithm and number of Bids are shown in Table 3 and Fig.4.

Table- 3 Runtime Analysis of Randomized QuickSort and Partial QuickSort

Bids	Execution Time(average time in seconds)	
	Randomized QuickSort	Partial QuickSort
1000	0.15	0.15
5000	0.24	0.22
10,000	0.40	0.36
20,000	0.46	0.43
30,000	0.51	0.47
40,000	0.60	0.56
50,000	0.62	0.59

In the below Fig.4, On the x-axis we have placed the Number of Bids and on the y-axis we have taken the execution time in seconds. The Execution Time includes the printing of sorted result.

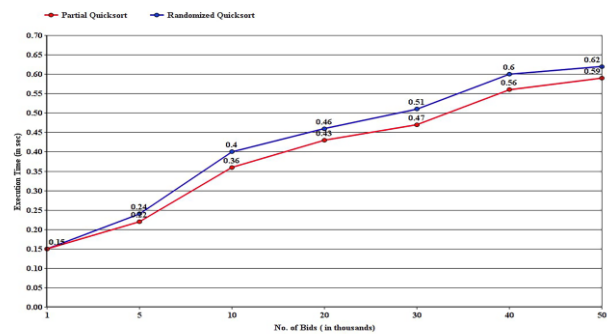


Fig. 4. This is a plot of Time Comparison of Randomized QuickSort(Blue) and Partial QuickSort (Red) over the random data generated by random number generator.

It shows that even when results are judged on the basis of random pivot used by both the sorting algorithms, The Partial QuickSort Algorithm turns out to be the best in both them. From the Comparison 1(Quick sort, Merge Sort with Partial Quick Sort) and Comparison 2(Randomized QuickSort with partial QuickSort)it is ruled out that Partial QuickSort is the best Sorting algorithm for Live Auction.



Partial Sort would result a list of largest bids of those concurrent bids which were put by the multiple users in the Live Auction. These Largest bids are now compared for their uniqueness. If there does not exist any redundancy, the highest bid is found. Otherwise, the time stamp of those redundant bids is taken into consideration for the result of highest bid i.e.; the bid having smallest timestamp will be considered as highest bid.

```

Void redundant(int[] A,int m)
{
for(i:m)
{
int first=A[highest]; int second;
if(A[highest]==A[i] )
{ second=A[i];
if(first.timestamp<second.timestamp )
{
A[highest]=first;
} else if(second.timestamp<first.timestamp)
{
A[highest]=second;
}
}
}
Else
Continue;
}}
    
```

**Algorithm of Redundancy Check**

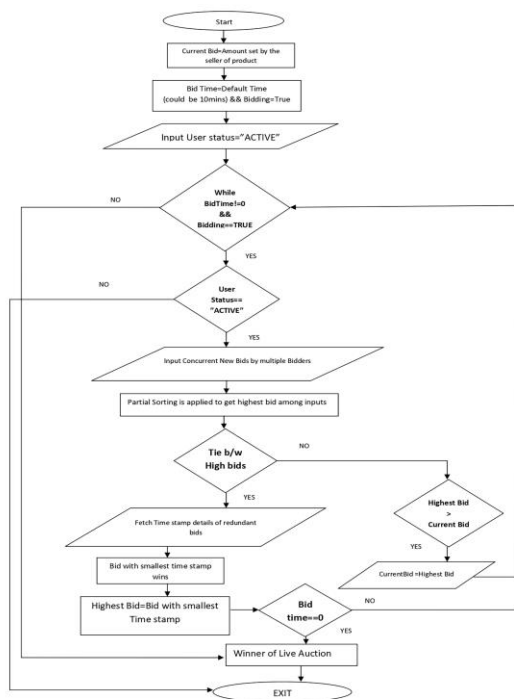


Fig. 5. The Flow of Working of Live Auction

**III. CONCLUSION**

The predicted algorithm is used in the bidding process to sort the bids done by the multiple bidders over any particular product. The basic aim of this algorithm is to reduce the unnecessary predicting pivot element randomly and sorting only necessary partition recursively. The sorted list of highest bids are then reviewed for redundancy check. A preliminary analysis of partial quick sort exhibits its competitive advantage over other alternative.

**IV. ACKNOWLEDGEMENT**

This paper is supported by the work of Conrado Mart Inez(2004) on partial Quicksort.

**V. REFERENCE**

- [1] Liu Mengjuan, Li Jiaying, Yue Wei , Liu Lizhou Qiu Jinyu, Qin Zhiguang (2019). „An Intelligent Bidding Strategy Based On Model-Free Reinforcement Learning For Real-Time Bidding In Display Advertising, Ieee 19211862,Doi 10.1109.
- [2] Liu Shuhao And Yu Yong (2019). Bid-Aware Active Learning In Real-Time Bidding For Display Advertising”, Digital Object Identifier 10.1109/Access.2019.2961155.
- [3] Karlsson Niklas (2019). Systems And Methods For Improved Sorting Using Intelligent Partitioning And Termination ,Patent Number 20190311425.
- [4] Tengyun Wang, Haizhi, Yang Han Yu, Zhou Wenjun, Yang Liu, And Song Hengjie,(2019). A Revenue-Maximizing Bidding Strategy For Demand-Side Platforms.DOI 10.1109/ACCESS.2019.2919450.
- [5] Sawant Geetanjali, Bane Ganesh , Gurav Akshay , Pawar Swaraj , (2018).Survey On Online Auction System ,Iosr Journal Of Computer Engineering (Iosr-Jce) E-Issn: 2278-0661,P-Issn: 2278-8727 Pp 58-60 .
- [6] Bhamare Manasi , Chame Arati , More Gaurav, Prof. Rindhe Amol , A Web-Based Online Auction System:A Survey, Doi:10.15680.
- [7] Zhang Jie , Li Linjing , And Wang Fei-Yue(2017). Probabilistic Mechanism Design For Online Auctions, Digital Object Identifier 10.1109/ACCESS.2017.2705120.
- [8] Chothani Rashesh G , Patel Nainesh A, Dekavadiya Asagarali H, Patel Punit R (2015). A Review Of Online Auction And It’s Pros And Cons ,International Journal Of Advance Engineering And Research Development (Ijaerd) E-Issn:2348 - 4470 .
- [9] Kumar M. N. and Singh M. R. (2014) ,Performance Comparison of Sorting Algorithms On The Basis Of Complexity, International Journal of Computer Science and Information Technology Research Vol. 2, no. 3, (pp. 394-398) .



- [10] Rauniar, Morefield ,Rupak, Roger D.,Simms, John ,Rauniar, Deepak(2009) ,Online Auctions: A Study Of Bidder Satisfaction. Volume 16 Number 1.
- [11] Durand Marianne (2005),Summary of Mart Inez Conrado Forty Years Of 'Quicksort' And 'Quickselect': A Personal View , Algorithms Seminar 2002–2004 .(pp. 101–104)
- [12] Mart Inez Conrado (2004) Partial Quicksort ,The Research Of The Author Was Supported By The Future And Emergent Technologies Programme Of *Issn 1999-14186*.