**ASSIGNMENT NO.**

**Title:** Concurrent ODD-EVEN SORT.

**Aim:** A web application for concurrent ODD-EVEN SORT.

**Objective:** Modeling,Testing and implementation of concurrent ODD-EVEN SORT.

**Theory:**

Sorting is a method which arranges the list of elements into a particular order. sorting has two different meanings ordering and categorizing, ordering means to order the list of same items and categorizing means grouping and labeling the same type of items. sorting is used in other algorithms that require sorted list to work efficiently. Odd even merge sort algorithm can be used for the construction of a systematic sorting network. To construct a systematic sorting network, it is necessary to construct a comparison network that can sort any odd-even sequence. The main idea of this algorithm is that first it sorts the odd position list and the even position list, finally it combines the two sorted list into a single sorted sequence by using a merge algorithm.

Initially we accept a single array from the user and then split into two, one of odd sequence and other of even sequence. Then each sequence is sorted in parallel by using merge sort, and then we get two array which are sorted. The two arrays are merged and sorted into a single array.

Partition(): This method is used for partitioning the odd and even array and then it call the merge sort for each partition.

Merge(): This method is used to combine the two sorted array that is even and odd and form the final single sorted array.

**Threading:**

Running several threads is similar to running several different programs concurrently, but with the following benefits −

* Multiple threads within a process share the same data space with the main thread and can therefore share information or communicate with each other more easily than if they were separate processes.
* Threads sometimes called light-weight processes and they do not require much memory overhead; they care cheaper than processes.

A thread has a beginning, an execution sequence, and a conclusion. It has an instruction pointer that keeps track of where within its context it is currently running.

* It can be pre-empted (interrupted)
* It can temporarily be put on hold (also known as sleeping) while other threads are running - this is called yielding.

The **methods** provided by the *Thread* class which are used are as follows:

**start():** The start() method starts a thread by calling the run method.

**join([time]):** The join() waits for threads to terminate.

**Software Modelling & designing:**

**Mathematical Model:**

Let S be the system such that:

S={s,e,X,Y,F,Sc,Fc}

Where,

s= initial state

e= end state

X= set of inputs

Y= set of outputs

F= set of function

Sc= Success cases

Fc= Failure cases

Let S’ be system in observation

Where S’ C S

S’ = {s,e,X,Y,F,Sc,Fc}

* S= start state

{count,init\_arr }

* e= end state

exit(0) ….success

* X= {(Xi,n) | XiЄ i, 0 ≤ i <n }
* Y= {Y1, Y2, Y3} Є Y

Where ,

{Y1,}Є success

{ Y2} Є failure

* F= {F1, F2,F3}

F1 = odd-even-sort(arr,n)

F2 = myThreads()

F3 = serverConnection()

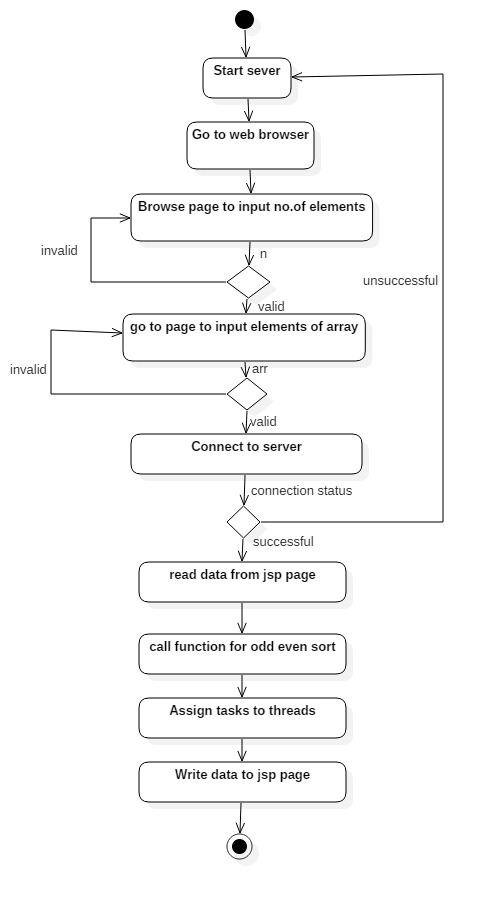
* Sc= {Y1,Y2}

where Y1 = {Xi<Xj| i<j<n and i != j}

* Fc = {Y3}

Where Y3 = { Xi>Xj| i<j<n and i != j }

**Activity Diagram:**

****

**Fig: Activity Diagram**

**System Testing:**

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

**White Box Testing Techniques**:

* Statement Coverage - This technique is aimed at exercising all programming statements with minimal tests.
* Branch Coverage - This technique is running a series of tests to ensure that all branches are tested at least once.
* Path Coverage - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

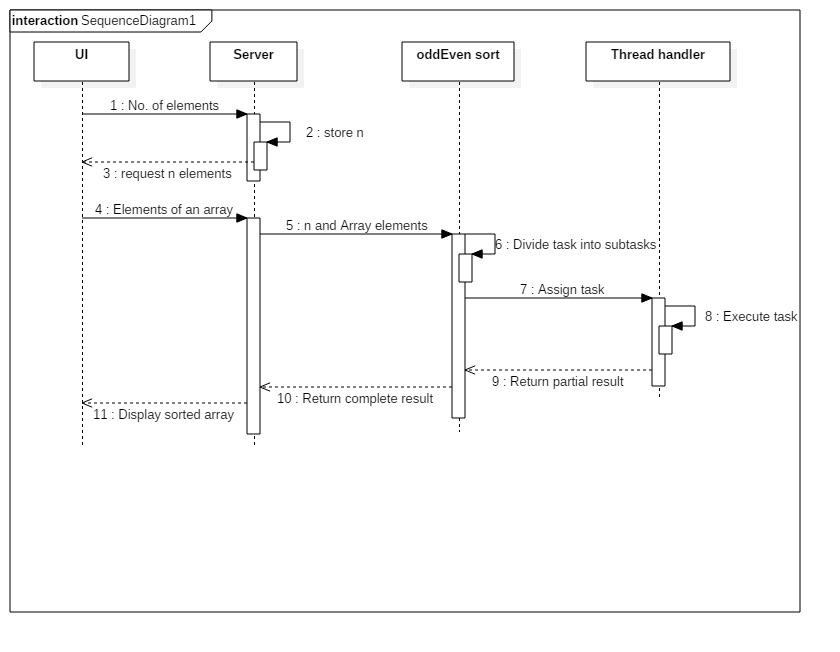
**Advantages of White Box Testing**:

* Forces test developer to reason carefully about implementation.
* Reveals errors in "hidden" code.
* Spots the Dead Code or other issues with respect to best programming practices.

**Disadvantages of White Box Testing:**

* Expensive as one has to spend both time and money to perform white box testing.
* Every possibility that few lines of code are missed accidentally.
* In-depth knowledge about the programming language is necessary to perform white box testing.

**Sequence Diagram:**

****

**Fig: Sequence Diagram**

**Algorithm**:

1. var sorted = false;
2. while list is not sorted
3. sorted = true; i=1
4. while i is less than n
5. if(list[i] > list[i+1])

swap(i, i+1);

sorted = false;

i=i+2

1. i=0
2. while i is less than n
3. if(list[i] > list[i+1])

swap(, i, i+1);

sorted = false;

i=i+2

**Input**: number of elements, array of elements

**Output**:Sorted array

**Platform**: Ubuntu 12.04

**Language**: Java

**Conclusion :**

        Hence, Wehave implemented and designed concurrent ODD-EVEN sort