

Report

Algorithm:

Input :

First of all, we input the size of an Array to manipulate it. It's size (this input) should be the power of 2. If this is a positive integer and in the power of 2, It will manipulate the tasks, otherwise it asks to Input again till condition satisfied.

Populate Array:

As we have size of an array so we declared an array of given size. Then call a function named "populate()" which argument are Array pointer and size of Array, to populate it by random numbers.

Splitting :

We are using Queue built-in Data Structure In this Algorithm.

We calculate mid and display it by calling a function Named "Display()" which argument are Array pointer ,starting index and ending index. Then we display it in two parts as left side from start till mid and right side from mid to end by calling same function with difference of indexes . Then store it these indexes(start , mid, mid+1,end) in a queue (Data structure Object) and pop a side from the queue (indexes) ,It considered as start and end of an Array, perform this function (display) on it and then store in the queue (object) in form of further splitting. We store indexes till start !=mid , and pop till queue not empty.

This Loop splits Array into two parts , display it and then stores indexes and get indexes for next operation.

Time Complexity:

To calculate the execution of the method, we use this kind of statements.

```
int start_t = clock(); //start of the method
int stop_t = clock();  // end of the method
double exeTime = (stop_t -
start_t)/double(CLOCKS_PER_SEC)*1000;
cout << "Execution Time: " << exeTime << endl;
```

Big Oh:

- Its first loop is for populating Array which executes in N time.
- It splits Array in N times.
- And when it Displays again and again.
Its Big Oh become N^2

$$O(N)=N^2$$

Memory:

Array Size = $N*4$ Bytes //4 Bytes is size of int

Variables = $1*4$ Bytes

Populate() function = $n*4$ Bytes

Display() function Max = $N*4$ Bytes

SplitArray() function variables = $5*4$ Bytes

Queue = $\log_2(n)*4 *4$ //first 4 is pushes for indexes

// 2nd 4 is size of int

Total Memory Used = sum of all above sizes

$$= N*4 + 1*4 + N*4 + N*4 + 5*4 + \log_2(N)*16$$

$$= 4(N + 6 + N + N + \log_2(N)*4)$$

$$= 4(3N+7+\log_2(N)*4) \text{ Bytes}$$