BINARY SERCH

**What is Binary Search?**

Binary search is a widely used algorithm for finding a specific element (target) in a sorted array or list. It works by repeatedly dividing the search interval in half and comparing the middle element of the current interval with the target element. Based on the comparison, the algorithm narrows down the search space until it finds the target element or determines that it doesn't exist in the array.

**History of searching and sorting an array.**

From the beginning of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement. Among the authors of early sorting algorithms around 1951 was Betty Holberton, who worked on ENIAC and UNIVAC.Bubble Sort was analyzed as early as 1956. Asymptotically optimal algorithms have been known since the mid-20th century – new algorithms are still being invented, with the widely used Timsort dating to 2002, and the library sort being first published in 2006.

**Different Implementation of Binary Search**

The Binary Search Algorithm can be implemented in the following two ways:

○ Iterative Binary Search Algorithm :- In it we basically use a function to return some values using loops and other functions.

○ Recursive Binary Search Algorithm :- In it we basically use a function to call itself and repeat it until the required conditions are fulfilled.

**Your Interpretation.**

#include<stdio.h>

int binary\_search(int arr[], int start, int end, int s);

void swap(int \*a, int \*b);

void selectionShort(int a[], int size);

int main() {

int size, s;

printf("Enter the size of array: ");

scanf("%d", &size);

int array[size];

printf("\nEnter the elements of array:\n");

for(int i=0; i<=end) {

int mid = start + (end - start)/2;

if(arr[mid] == s) {

return mid;

}

if(arr[mid] < s) {

start = mid + 1;

}

else {

end = mid - 1;

}

}

return -1;

}

void swap(int \*a, int \*b) {

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

void selectionShort(int a[], int size) {

int min\_index;

for(int i=0; i< a[min\_index]) {

min\_index = i;

for(int j = i+1; j< a[min\_index]) {

if(a[j] < a[min\_index]) {

min\_index = j;

}

}

if(a[i] > a[min\_index]) {

swap(&a[i], &a[min\_index]);

}

}

}