

## ESc101 : Practice Sheet 2

Ques 1: Binary search is an efficient algorithm to search a given element  $x$  in a sorted array of  $n$  elements.

The idea of binary search is to use the information that the array is sorted. We basically ignore half of the elements just after one comparison.

The algorithm for Binary Search is as follows -

- 1) Compare  $x$  with the middle element.
- 2) If  $x$  matches with middle element, we return the mid index.
- 3) Else If  $x$  is greater than the mid element, then  $x$  can only lie in right half subarray after the mid element. So we search in the right half.
- 4) Else ( $x$  is smaller) search in the left half.
- 5) Return -1, if element is not present in the array.

Below is the code for binary search. Complete it for the given algorithm -

```
int binarySearch(int arr[], int l, int r, int x)
{
    while (l <= r)
    {
        int m = l + (r-l)/2;

        if (arr[m] == x) return ( _ _ _ _ );

        if ( _ _ _ _ ) l = m + 1;

        else r = ( _ _ _ _ );
    }
    return -1; // if we reach here, then element was not present
}

int main(void)
{
    int arr[] = {2, 3, 4, 10, 40};
    int n = sizeof(arr)/ sizeof(arr[0]);
    int x = 10;
    int result = binarySearch(_ _ _ _ );
    (result == -1)? printf("Element is not present in array")
                  : printf("Element is present at index %d", result);
    return 0;
}
```

Ques 2 : Merge algo takes two sorted arrays as input and merge them in another array such that the resulting array remains sorted.

Input : sorted array a[] and b[] with m and n elements respectively.

Output : sorted array sorted[] of size m+n.

Using this information, complete the following code -

```
void merge(int a[], int m, int b[], int n, int sorted[]) {
    int i, j, k;
    j = k = 0;
    for (i = 0; i < m + n; i++) {
        if (a[j] < b[k]) {
            sorted[i] = a[j];
            j++;
        }
        else {
            sorted[i] = b[k];
            k++;
        }
    }
    else if (j == m) {
        for (; i < m + n; i++) {
            sorted[i] = b[k];
            k++;
        }
    }
    else {
        for (; i < m + n; i++) {
            sorted[i] = a[j];
            j++;
        }
    }
}
```