Bahria University,

Karachi Campus



COURSE: CSC-221 DATA STRUCTURES AND ALGORITHM

TERM: FALL 2020, CLASS: BSE- 3 (A)

Submitted By:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ADIL WAHEED) (65190)

Enrollment #:02-131192-082

Submitted To:

Engr. Dr. Farah/ Engr. Ramshaa

Signed Remarks: Score:

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
| 01 | 1-10-2020 | 01 | ONE AND TWO DIMENSIONAL ARRAY |  |
| 02 | 09-10-20 | 02 | Linear Search & Sorting Algorithms |  |
| 03 | 13-10-20 | 03 | Recusrion |  |
| 04 | 30/10/2020 | 04 | Binary Search Algorithm |  |
| 05 | 30/10/20 | 05 | Merge Sort |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_05\_\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Implement merge sort algorithm to merge two character arrays into a third array in sorted order. |
| 2 | Implement merge sort on following array [25,98,1,14,3] and search 14 value on sorted array using any searching algorithm |
|  |  |
|  |  |
|  |  |

Submitted On:

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: 31/10/20)

**Task No. 1: Implement merge sort algorithm to merge two character arrays into a third array in sorted order.**

**Solution:**

static char[] merge(char[] left, char[] right, char[] array)

{

int nl, nr, i, j, k;

nl = nr = i = j = k = 0;

nl = left.Length;

nr = right.Length;

while (i < nl && j < nr)

{

if (left[i] <= right[j])

{

array[k] = left[i];

k++;

i++;

}

else

{

array[k] = right[j];

k++;

j++;

}

}

while (i < nl)

{

array[k] = left[i];

k++;

i++;

}

while (j < nr)

{

array[k] = right[j];

k++;

j++;

}

return array;

}

static char[] mergesort(char[] array)

{

int n = array.Length;

if (n < 2)

return null;

int mid = n / 2;

char[] arrayleft = new char[mid];

char[] arrayright = new char[n - mid];

for (int i = 0; i <= mid - 1; i++)

{

arrayleft[i] = array[i];

}

for (int i = mid; i <= n - 1; i++)

{

arrayright[i - mid] = array[i];

}

mergesort(arrayleft);

mergesort(arrayright);

return merge(arrayleft, arrayright, array);

}

static void Main(string[] args)

{

char[] a = new char[] { 'a', 'd', 'r', 'u' };

char[] b = new char[] { 'v', 'b', 'x', 'z' };

char[] c = new char[a.Length + b.Length];

//int[] a = new int[] { 7, 8, 4, 5, 6 };

//int[] b = new int[] { 1,2,9,3 };

//int[] c = new int[a.Length + b.Length];

Console.WriteLine("Array A");

foreach (char arrA in a)

{

Console.Write(arrA + " ");

}

Console.WriteLine("\n\nArray B");

foreach (char arrB in b)

{

Console.Write(arrB + " ");

}

Console.WriteLine();

a = mergesort(a);

b = mergesort(b);

c = merge(a,b,c);

Console.WriteLine("\nAfter Merging A and B");

foreach (char item in c)

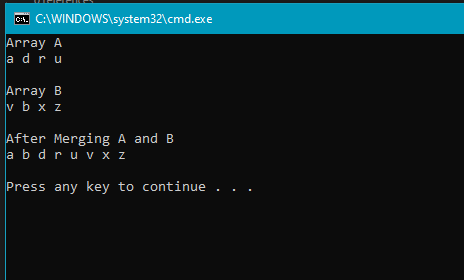
{

Console.Write(item + " ");

}

Console.WriteLine("\n");

**OUTPUT**:



**Task No. 2: Implement merge sort on following array [25,98,1,14,3] and search 14 value on sorted array using any searching algorithm**

**Solution:**

static int[] merge(int[] left, int[] right, int[] array)

{

int nl, nr, i, j, k;

nl = nr = i = j = k = 0;

nl = left.Length;

nr = right.Length;

while (i < nl && j < nr)

{

if (left[i] <= right[j])

{

array[k] = left[i];

k++;

i++;

}

else

{

array[k] = right[j];

k++;

j++;

}

}

while (i < nl)

{

array[k] = left[i];

k++;

i++;

}

while (j < nr)

{

array[k] = right[j];

k++;

j++;

}

return array;

}

static int[] mergesort(int[] array)

{

int n = array.Length;

if (n < 2)

return null;

int mid = n / 2;

int[] arrayleft = new int[mid];

int[] arrayright = new int[n - mid];

for (int i = 0; i <= mid - 1; i++)

{

arrayleft[i] = array[i];

}

for (int i = mid; i <= n - 1; i++)

{

arrayright[i - mid] = array[i];

}

mergesort(arrayleft);

mergesort(arrayright);

return merge(arrayleft, arrayright, array);

}

public static int search(int[] arr, int x)

{

int n = arr.Length;

for (int i = 0; i < n; i++)

{

if (arr[i] == x)

return i;

}

return -1;

}

static void Main(string[] args)

{

int[] a = new int[] { 25, 98, 1, 14, 3 };

Console.WriteLine("Before merge sort");

foreach (var item in a)

{

Console.Write(" "+item);

}

Console.WriteLine();

a= mergesort(a);

Console.WriteLine("After merge sort");

foreach (var item in a)

{

Console.Write(" " + item);

}

Console.WriteLine();

Console.WriteLine("Applying Linear Search");

int result = search(a,14);

if (result == -1)

Console.WriteLine(

"Element is not present in array");

else

Console.WriteLine("Element is present at index "

+ result);

}

}

**OUTPUT**:

