Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**07**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **Implement recursive method of merge sort algorithm to sort an array of 10 characters.** |
| **2** | **Implement quick sort algorithm on string array using left value as first pivoting value.** |
|  |  |
|  |  |

Submitted On:

**23-12-21**

(Date: DD/MM/YY)

**Task 1: Implement recursive method of merge sort algorithm to sort an array of 10 characters.**

**SOLUTION:** class MergeSort

{ void merge(char[] arr, int l, int m, int r)

{ int n1 = m - l + 1;

int n2 = r - m;

char[] L = new char[n1];

char[] R = new char[n2];

int i, j;

for (i = 0; i <= n1 - 1; i++)

L[i] = arr[l + i];

for (j = 0; j <= n2 - 1; j++)

R[j] = arr[m + 1 + j];

i = 0;

j = 0;

int k = l;

while (i < n1 && j < n2)

{ if (L[i] <= R[j])

{ arr[k] = L[i];

i++; }

else

{arr[k] = R[j];

j++; }

k++;}

while (i < n1)

{arr[k] = L[i];

i++;

k++;}

while (j < n2)

{arr[k] = R[j];

j++;

k++;} }

void sort(char[] arr, int l, int r)

{ if (l < r)

{ int m = l + (r - l) / 2;

sort(arr, l, m);

sort(arr, m + 1, r);

merge(arr, l, m, r); } }

static void printArray(char[] arr)

{for (int i = 0; i < arr.Length; i++)

Console.Write(arr[i] + " ");

Console.WriteLine();}

public static void Main(String[] args)

{char[] arr = { 'J', 'M', 'F', 'C', 'A', 'Z', 'G', 'I', 'L', 'V' };

Console.WriteLine("Given Array ");

printArray(arr);

MergeSort obj = new MergeSort();

obj.sort(arr, 0, arr.Length - 1);

Console.WriteLine("\nSorted array ");

printArray(arr);

**OUTPUT:**

Text

Description automatically generated

**Task 2: Implement quick sort algorithm on string array using left value as first pivoting value.**

**SOLUTION:**

public class QuickSort

{ public void quickSort(string[] arr, int left, int right)

{ int l = left;

int r = right;

String pivot = arr[left];

while (l <= r)

{ while (arr[l].ToLower().CompareTo(pivot.ToLower()) < 0)

l++;  
 while (arr[r].ToLower().CompareTo(pivot.ToLower()) > 0)

r--;

if (l <= r)

{ String temp = arr[l];

arr[l] = arr[r];

arr[r] = temp;

l++;

r--; }}

if (left < r)

quickSort(arr, left, r);

if (l < right)

quickSort(arr, l, right); }}

class Program

{ public static void printArray(string[] arr)

{ for (int i = 0; i <= arr.Length - 1; i++)

Console.Write(arr[i] + " ");

Console.WriteLine();}

static void Main(string[] args)

{

string[] arr = { "HANNAN", "NIKE", "HAMZA", "DAVID", "ALY" };

Console.WriteLine("Given Array");

printArray(arr);

QuickSort obj = new QuickSort();

obj.quickSort(arr, 0, arr.Length - 1);

Console.WriteLine("\nSorted array");

printArray(arr);

**OUTPUT:**

Text

Description automatically generated