

DAA Practical No: 05

Q) Write a C/C++ code to implement (with practical example implementation)

- 1) Binary Search
- 2) Merge Sort
- 3) Quick Sort
- 4) Strassen's Matrix multiplication

1) Binary Search

-- > Program Code:

```
Go Run Terminal Help BinarySearch.cpp - DAA - Visual Studio Code

BinarySearch.cpp X
p5 > BinarySearch.cpp > binary(int [], int, int)
1 //Binary Search
2 #include<iostream>
3 using namespace std;
4
5 int binary(int arr[],int size,int key ){
6     int start=0;
7     int end=size-1;
8
9     // int mid= (start + end)/2; //if two big values of start and end we take
10    // then it may go out of range of int
11    int mid= start + (end-start)/2;
12
13
14    while(start <= end){
15
16        if(arr[mid] == key){
17            return mid;
18        }
19
20        //to go right part
21        else if(arr[mid]> key){
22            start=mid+1;
23        }
24
25        //to go left part
26        else {
27            end = mid-1;
28        }
29
30        mid= start + (end-start)/2;
31    }
32    return -1;
33
34 }
35
```

```

34
35 int main(){
36     int arr[5]={1,2,5,10,20};
37     int brr[6]={20,10,8,7,5,3};
38
39     int key;
40     cout<<"Enter number to search in array:"<<endl;
41     cin>>key;
42
43
44     cout<<"Index of key in oddlength array is:"<<binary(arr,5,key)<<endl;
45     cout<<"index of key in evenlength array is:"<<binary(brr,6,key)<<endl;
46
47 }

```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

● PS C:\Users\DELL\Desktop\c++> cd "c:\Users\DELL\Desktop\c++"
● PS C:\Users\DELL\Desktop\c++> & .\"22binarySearch.exe"
Enter number to search in array:
○ 5
Index of key in oddlength array is:2
index of key in evenlength array is:4
PS C:\Users\DELL\Desktop\c++>

```

2) Merge Sort:

--> Program Code:

```
Go Run Terminal Help mergeSort.cpp - DAA - Visual Studio Code

mergeSort.cpp X
p5 > mergeSort.cpp > Merge(int *, int, int, int)
1  #include <iostream>
2  using namespace std;
3
4  // A function to merge the two half into a sorted data.
5  void Merge(int *a, int low, int high, int mid)
6  {
7      // We have low to mid and mid+1 to high already sorted.
8      int i, j, k, temp[high-low+1];
9      i = low;
10     k = 0;
11     j = mid + 1;
12     while (i <= mid && j <= high){        // Merge the two parts into temp[].
13         if (a[i] < a[j]){
14             temp[k] = a[i];
15             k++;
16             i++;}
17         else{
18             temp[k] = a[j];
19             k++;
20             j++;}
21     }
22     // Insert all the remaining values from i to mid into temp[].
23     while (i <= mid){
24         temp[k] = a[i];
25         k++;
26         i++;}
27     // Insert all the remaining values from j to high into temp[].
28     while (j <= high){
29         temp[k] = a[j];
30         k++;
31         j++;}
32     // Assign sorted data stored in temp[] to a[].
33     for (i = low; i <= high; i++){
34         a[i] = temp[i-low];
35     }
36 }
37
```

```
Go Run Terminal Help mergeSort.cpp - DAA - Visual Studio Code
mergeSort.cpp X
p5 > mergeSort.cpp > Merge(int *, int, int, int)
37
38 // A function to split array into two parts.
39 void MergeSort(int *a, int low, int high)
40 {
41     int mid;
42     if (low < high){
43         mid=(low+high)/2;
44         // Split the data into two half.
45         MergeSort(a, low, mid);
46         MergeSort(a, mid+1, high);
47
48         // Merge them to get sorted output.
49         Merge(a, low, high, mid);
50     }
51 }
52
53 int main(){
54     int n, i;
55     cout<<"\nEnter the number of data element to be sorted: ";
56     cin>>n;
57
58     int arr[n];
59     for(i = 0; i < n; i++){
60         cout<<"Enter element "<<i+1<<": ";
61         cin>>arr[i];
62     }
63
64     MergeSort(arr, 0, n-1);
65
66     // Printing the sorted data.
67     cout<<"\nSorted Data ";
68     for (i = 0; i < n; i++)
69         cout<<"->"<<arr[i];
70
71     return 0;
72 }
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
● PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5"
● PS C:\Users\DELL\Desktop\DAA\p5> & .\"mergeSort.exe"

Enter the number of data element to be sorted: 5
Enter element 1: 20
Enter element 2: 45
Enter element 3: 85
Enter element 4: 10
Enter element 5: 0

Sorted Data ->0->10->20->45->85
○ PS C:\Users\DELL\Desktop\DAA\p5> █
```

3)Quick Sort:

-- > Program Code:

Go Run Terminal Help quickSort.cpp - DAA - Visual Studio Code

quickSort.cpp X

```
p5 > quickSort.cpp > ...
1 //Quick Sort
2 #include<iostream>
3 using namespace std;
4
5 void swap(int arr[] , int pos1, int pos2){
6     int temp;
7     temp = arr[pos1];
8     arr[pos1] = arr[pos2];
9     arr[pos2] = temp;
10 }
11
12 int partition(int arr[], int low, int high, int pivot){
13     int i = low;
14     int j = low;
15     while( i <= high){
16         if(arr[i] > pivot){
17             i++;
18         }
19         else{
20             swap(arr,i,j);
21             i++;
22             j++;
23         }
24     }
25     return j-1;
26 }
27
28 void quickSort(int arr[], int low, int high){
29     if(low < high){
30         int pivot = arr[high];
31         int pos = partition(arr, low, high, pivot);
32
33         quickSort(arr, low, pos-1);
34         quickSort(arr, pos+1, high);
35     }
36 }
37
```

```
38 int main()
39 {
40     int n ;
41     cout << " enter the size of array: ";
42     cin>>n;
43     int arr[n];
44     cout<<"Enter elements in array:";
45     for( int i = 0 ; i < n; i++){
46         cin>> arr[i];
47     }
48     quickSort(arr, 0 , n-1);
49     cout<<"The sorted array is: "<<endl;
50     for( int i = 0 ; i < n; i++){
51         cout<< arr[i]<<" ";
52     }
53
54 }
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

● PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5\output"
● PS C:\Users\DELL\Desktop\DAA\p5\output> & .\"quickSort.exe"
  enter the size of array: 5
  Enter elements in array:45 10 20 80 70
  The sorted array is:
  10 20 45 70 80
○ PS C:\Users\DELL\Desktop\DAA\p5\output> █
```

4)Strassen's Matrix multiplication

Program Code:

```
Go Run Terminal Help    matrix_multiplication.cpp - DAA - Visual Studio Code

matrix_multiplication.cpp X
p5 > matrix_multiplication.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  void multiply(int[5][5], int[5][5], int, int, int);
5  int display(int[5][5], int, int);
6
7  int main()
8  {
9      int a[5][5], b[5][5], r1, c1, r2, c2;
10     cout << "\n Enter rows for first matrix: ";
11     cin >> r1;
12     cout << "\n Enter columns for first matrix: ";
13     cin >> c1;
14
15     cout << "\n Enter rows for second matrix: ";
16     cin >> r2;
17     cout << "\n Enter columns for second matrix: ";
18     cin >> c2;
19
20     // To check if columns of first matrix are equal to rows of second matrix
21     if (c1 != r2)
22     |     return 0;
23     // Storing elements of first matrix.
24
25     cout << "\n Enter elements of first matrix \n";
26
27     for (int i = 0; i < r1; i++) {
28     |     for (int j = 0; j < c1; j++)
29     |     |     cin >> a[i][j];
30     |     }
31     // Storing elements of second matrix.
32     cout << "\n Enter elements of second matrix\n";
33
34     for (int i = 0; i < r2; i++) {
35     |     for (int j = 0; j < c2; j++)
36     |     |     cin >> b[i][j];
37     |     }
```

```

38     display(a, r1, c1);
39     display(b, r2, c2);
40     //calling the function to multiply a and b. passing number of rows
41     //and columns in both of them
42     multiply(a, b, r1, c2, c1);
43     return 0;
44 }
45 void multiply(int a[5][5], int b[5][5], int row, int col, int c1)
46 {
47     int c[5][5];
48     //input 0 for all values of c, in order to remove
49     //the garbage values assigned earlier
50     for (int i = 0; i < row; i++) {
51         for (int j = 0; j < col; j++)
52             c[i][j] = 0;
53     }
54     //we apply the same formula as above
55     for (int i = 0; i < row; i++) {
56         for (int j = 0; j < col; j++) {
57             for (int k = 0; k < c1; k++) //columns of first matrix || rows of second matrix
58                 c[i][j] += a[i][k] * b[k][j];
59         }
60     }
61     cout << "\n Matrix c after matrix multiplication is:\n";
62     display(c, row, col);
63 }
64 int display(int c[5][5], int row, int col)
65 {
66     cout << "\n Matrix is:\n";
67     for (int i = 0; i < row; i++) {
68         for (int j = 0; j < col; j++)
69             cout << c[i][j] << " ";
70         cout << "\n";
71     }
72     return 0;
73 }

```

Output:

Go Run Terminal Help matrix_multiplication.cpp - DAA - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

● PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5\output"
● PS C:\Users\DELL\Desktop\DAA\p5\output> & .\"matrix_multiplication.exe"

Enter rows for first matrix: 2

Enter columns for first matrix: 3

Enter rows for second matrix: 3

Enter columns for second matrix: 2

Enter elements of first matrix
5 8 9
1 3 7

Enter elements of second matrix
5 2
10 4
7 3

Matrix is:
5 8 9
1 3 7

Matrix is:
5 2
10 4
7 3

Matrix c after matrix multiplication is:

Matrix is:
168 69
84 35
PS C:\Users\DELL\Desktop\DAA\p5\output>

```