

Practical No: 3

Name: sanket pradiprao gaikwad

Reg.no : 2020BIT036

1) Searching Algorithm

a) Linear search

```
sanket.cpp > ...
1  #include<iostream>
2  using namespace std;
3  int linearSearch(int arr[], int n, int target){
4      for(int i=0; i<n; i++){
5          if(arr[i]==target){
6              return i;
7          }
8      }
9      return -1;
10 }
11 int main(){
12     int arr[]={5,6,7,8,9,14,17};
13     int n = 7;
14     cout<<linearSearch(arr,n,14);
15     return 0;
16 }
17 |
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```
PS D:\CODING\Programming> cd "d:\CODING\Programming\" ; if ($?) { g++ sanket.cpp -o sanket } ; if ($?) { .\sanket }
5
```

b) Binary search:

```
sanket.cpp > binarySearch(int [], int, int)
1  #include <iostream>
2  using namespace std;
3  int binarySearch(int arr[], int n, int target){
4      int l = 0;
5      int h = n - 1;
6      while (l <= h) {
7          int middle = (l + h) / 2;
8
9          if (arr[middle] == target) {
10             return middle;
11         }
12         else if (arr[middle] > target) {
13             h = middle - 1;
14         }
15         else{
16             l = middle + 1;
17         }
18     }
19     return -1;
20 }
21 int main(){
22     int arr[] = {1,3,4,5,6,7,8};
23     cout<< binarySearch(arr, 7, 4);
24     return 0;
25 }
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```
PS D:\CODING\Programming> cd "d:\CODING\Programming\" ; if ($?) { g++ sanket.cpp -o sanket } ; if ($?) { .\sanket }
2
```

c) Recursive Search:

```
sanket.cpp > recursive(int [], int, int, int)
1  #include<iostream>
2  using namespace std;
3  int recursive(int arr[], int low, int high, int target){
4      if(low>high){return -1;}
5      int mid = low + (high-low)/2;
6      if(arr[mid]==target){
7          return mid;
8      }
9      if(arr[mid]>target){
10         return recursive(arr, low, mid-1,target);
11     }
12     return recursive(arr, mid+1, high, target);
13 }
14 int main(){
15     int arr[]={1,3,4,5,6,8};
16     cout<<recursive(arr,0, 5,6);
17 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS D:\CODING\Programming> cd "d:\CODING\Programming\" ; if ($?) { g++ sanket.cpp -o sanket } ; if ($?) { .\sanket }
4
```

2) Sorting Algorithm:

a) Insertion sort:

```
sanket.cpp > insertionSort(int [], int)
1  #include<bits/stdc++.h>
2  void insertionSort(int arr[], int n){
3      int i, key, j;
4      for (i = 1; i < n; i++) {
5          key = arr[i];
6          j = i - 1;
7          while (j >= 0 && arr[j] > key) {
8              arr[j + 1] = arr[j];
9              j = j - 1;
10         }
11         arr[j + 1] = key;
12     }
13 }
14 int main(){
15     int arr[] = { 12, 11, 13, 5, 6 };
16     int n = sizeof(arr) / sizeof(arr[0]);
17     insertionSort(arr, n);
18     for (int i = 0; i < n; i++)
19         printf("%d ", arr[i]);
20     return 0;
21 }
22
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS D:\CODING\Programming> cd "d:\CODING\Programming\" ; if ($?) { g++ sanket.cpp -o sanket } ; if ($?) { .\sanket }
5 6 11 12 13
```

b) Selection sort:

```
sanket.cpp > main()
1  #include <bits/stdc++.h>
2  using namespace std;
3  void selectionSort(int arr[], int n){
4      int i, j, min_idx;
5      for (i = 0; i < n-1; i++)
6      {
7          min_idx = i;
8          for (j = i+1; j < n; j++)
9              if (arr[j] < arr[min_idx])
10                 min_idx = j;
11             if(min_idx != i)
12                 swap(arr[min_idx], arr[i]);
13         }
14     }
15     int main(){
16         int arr[] = {64, 25, 12, 22, 11};
17         int n = sizeof(arr)/sizeof(arr[0]);
18         selectionSort(arr, n);
19         printf("Sorted array: \n");
20         for (int i=0; i < n; i++)
21             printf("%d ", arr[i]);
22         return 0;

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Sorted array:
11 12 22 25 64

c) Bubble Sort:

```
sanket.cpp > printArray(int [], int)
1  #include<bits/stdc++.h>
2  using namespace std;
3  void bubbleSort(int arr[], int n){
4      int i, j;
5      for (i = 0; i < n - 1; i++)
6          for (j = 0; j < n - i - 1; j++)
7              if (arr[j] > arr[j + 1])
8                  swap(arr[j], arr[j + 1]);
9  }
10 void printArray(int arr[], int size){
11     int i;
12     for (i = 0; i < size; i++)
13         printf("%d ", arr[i]);
14     printf("\n");
15 }
16 int main(){
17     int arr[] = { 5, 1, 4, 2, 8 };
18     int n = sizeof(arr) / sizeof(arr[0]);
19     bubbleSort(arr, n);
20     printf("Sorted array: \n");
21     printArray(arr, n);
22     return 0;

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Sorted array:
1 2 4 5 8
PS D:\CODING\Programming>