**Design and Analysis of Algorithm**

**Lab Record**

**Name:** Naveen Malhotra

**Reg No:** 209303050

**Branch:** B.Tech Computer and Communication Eng.

**Sem :** 5th

**Lab1:**

**Binary Search algorithm**

1. : Iterative method :-

#include <stdio.h>

int binarySearch(int array[], int key, int l, int h) {

while (l <= h) {

int mid = (l+h)/2;

if (array[mid] == key)

return mid;

if (array[mid] < key)

l = mid + 1;

else

h = mid - 1;

}

}

int main() {

printf("\n\n209303050\n\n");

int array[] = {1,4,7,9,12};

int n=5;

int key = 7;

int key2 = 9;

int result = binarySearch(array, key, 1,5);

int result2 = binarySearch(array, key2, 1,5);

printf("Element found at -> %d\n",result);

printf("Element found at -> %d\n\n",result2);

}

**Output:**

**Text

Description automatically generated**

**(b): Recursive method :-**

#include <stdio.h>

int binarySearch(int array[], int key, int l, int h) {

if (h >= l) {

int mid = l + (h - l) / 2;

if (array[mid] == key)

return mid;

if (array[mid] > key)

return binarySearch(array, key, l, mid - 1);

return binarySearch(array, key, mid + 1, h);

}

}

int main() {

printf("\n\n209303050\n\n");

int array[] = {1,4,7,9,12};

int n=5;

int key = 7;

int key2 = 9;

int result = binarySearch(array, key, 1,5);

int result2 = binarySearch(array, key2, 1,5);

printf("Element found at -> %d\n",result);

printf("Element found at -> %d\n\n",result2);

}

**Output:**

**Text

Description automatically generated**

**Lab 2:**

* **Quick Sort Algorithm:**

#include<stdio.h>

int quicksort(int number[25],int first,int last){

int i, j, pivot, temp;

if(first<last){

pivot=first;

i=first;

j=last;

while(i<j){

while(number[i]<=number[pivot])

i++;

while(number[j]>number[pivot])

j--;

if(i<j){

temp=number[i];

number[i]=number[j];

number[j]=temp;

}

}

temp=number[pivot];

number[pivot]=number[j];

number[j]=temp;

quicksort(number,first,j-1);

quicksort(number,j+1,last);

}

}

int main(){

int i, count, number[25];

printf("\n\nNaveen Malhotra(209303050)\n\n");

printf("Enter the no of elements in array: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

for(i=0;i<count;i++)

scanf("%d",&number[i]);

quicksort(number,0,count-1);

printf("Order of Sorted elements: ");

for(i=0;i<count;i++)

printf(" %d",number[i]);

printf("\n");

return 0;

}

**Output:**

**Text

Description automatically generated**

* **Bubble Sort Algorithm:**

#include<stdio.h>

#include<stdlib.h>

int main(){

printf("\n\nNaveen Malhotra (209303050) \n\n");

printf("Bubble Sort \n\n");

int arr[5]={18,6,1,5,4};

printf("Given Array -> ");

for(int i=0;i<5;i++){

printf("%d ",arr[i]);

}

printf("\n");

int temp;

for(int i=0;i<5;i++){

for(int j=0;j<4;j++){

if(arr[j]>arr[j+1]){

temp=arr[j];

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

arr[j+1]=temp;

}

}

}

printf("Sorted Array -> ");

for(int i=0;i<5;i++){

printf("%d ",arr[i]);

}

printf("\n\n");

}

**Output:**

**Text

Description automatically generated**

* **Selection Sort Algorithm:**

#include<stdio.h>

#include<stdlib.h>

int main(){

printf("\n\nNaveen Malhotra (209303050)\n\n");

int arr[5]={4,9,6,1,7};

int min

int temp=0;

printf("Given array is -> ");

for(int i=0;i<=4;i++){

printf("%d ",arr[i]);

}

printf("\n");

for(int i=0;i<5;i++){

min=i;

for(int j=i+1;j<5;j

if(arr[j]<arr[min])

{

min=j;

}

}

if(min != i){

temp=arr[min];

arr[min]=arr[i];

arr[i]=temp;

}

}

printf("Sorted array is -> ");

for(int i=0;i<=4;i++){

printf("%d ",arr[i]);

}

printf("\n");

}

**Output:**

**Graphical user interface, text

Description automatically generated**

**Lab 3:**

* **Merge Sort:**

#include <stdio.h>

#define max 10

int a[11];

int b[10];

void merging(int low, int mid, int high) {

int l1, l2, i;

for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) {

if(a[l1] <= a[l2])

b[i] = a[l1++];

else

b[i] = a[l2++];

}

while(l1 <= mid)

b[i++] = a[l1++];

while(l2 <= high)

b[i++] = a[l2++];

for(i = low; i <= high; i++)

a[i] = b[i];

}

void sort(int low, int high) {

int mid;

if(low < high) {

mid = (low + high) / 2;

sort(low, mid);

sort(mid+1, high);

merging(low, mid, high);

} else {

return;

}

}

int main() {

int i;

int n;

printf("\n\n NAVEEN MALHOTRA 209303050\n\n");

printf("Enter the size of array MAX 10 -> ");

scanf("%d",&n);

printf("Enter the array elements (MAX 10) -> ");

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

scanf("%d",&a[i]);

}

printf("\n");

printf("List before sorting\n");

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

printf("%d ", a[i]);

sort(0, n);

printf("\nList after sorting\n");

for(i = 1; i <= n; i++)

printf("%d ", a[i]);

printf("\n");

}

**OUTPUT:**

Text

Description automatically generated with low confidence

**Lab 4:**

**Knapsack Problem:**

#include<stdio.h>

void knapsack(int n, float weight[], float profit[], float capacity) {

float x[20], tp = 0;

int i, j, u;

u = capacity;

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

x[i] = 0.0; //

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

if(weight[i] > u)

break;

else {

x[i] = 1.0;

tp = tp + profit[i];

u = u - weight[i];

}

}

if(i < n)

x[i] = u / weight[i];

tp = tp + (x[i] \* profit[i]);

printf("Maximum profit is: %f\n", tp);

}

int main() {

float weight[20], profit[20], capacity;

int num, i, j;

float ratio[20], temp;

printf("\n\n Naveen Malhotra\n\n");

printf("Enter the no. of Items: ");

scanf("%d", &num);

printf("Enter the wts and profits of each object: ");

for(i = 0; i < num; i++) {

scanf("%f %f", &weight[i], &profit[i]);

}

printf("Enter the capacityacity of knapsack: ");

scanf("%f", &capacity);

for(i = 0; i < num; i++) {

ratio[i] = profit[i] / weight[i];

}

// sorting the ratio array in descending order using bubble sort algorithm

for(i = 0; i < num; i++) {

for(j = i + 1; j < num; j++) {

if(ratio[i] < ratio[j]) {

temp = ratio[j];

ratio[j] = ratio[i];

ratio[i] = temp;

temp = weight[j];

weight[j] = weight[i];

weight[i] = temp;

temp = profit[j];

profit[j] = profit[i];

profit[i] = temp;

}

}

}

knapsack(num, weight, profit, capacity);

return(0);

}

**OUTPUT:**

Text

Description automatically generated with medium confidence