**Week 1:**

**Question 1 :** Given an array of nonnegative integers, design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = O(n), where n is the size of input)

#include <iostream>

using namespace std;

int main()

{

int ch;

cout<<"enter the number of test cases"<<endl;

cin>>ch;

while(ch--)

{

int n,i,key,ctr=0;

cout<<"enter the number of elements"<<endl;

cin>>n;

int arr[n];

cout<<"enter the array elements"<<endl;

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

cin>>arr[i];

cout<<"enter the key"<<endl;

cin>>key;

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

{

if(arr[i]==key)

{ctr=1;

break;}

}

if(ctr==1)

cout<<"present "<<i+1<<endl;

else

cout<<"not present"<<i<<endl;

}

return 0;

}

**Output:**

enter the number of test cases

3

enter the number of elements

8

enter the array elements

34 35 65 31 25 89 64 30

enter the key

89

present 6

enter the number of elements

5

enter the array elements

977 354 244 546 355

enter the key

244

present 3

enter the number of elements

6

enter the array elements

23 64 13 67 43 56

enter the key

63

not present 6

Process returned 0 (0x0) execution time : 70.538 s

Press any key to continue.

**Question 2:** Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = O(nlogn), where n is the size of input).

#include<iostream>

using namespace std;

void linear\_search(int arr[],int n,int key)

{

int comparisions=0,flag=0;

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

{

comparisions++;

if (arr[i]==key)

{

cout<<"Present"<<" "<<comparisions<<endl;

flag=1;

break;

}

}

if (flag==0)

cout<<"Not Present"<<" "<<comparisions<<endl;

}

int main()

{

int t;

cout<<"enter the number of test cases"<<endl;

cin>>t;

while (t--)

{

int n;

cout<<"enter the number of elements"<<endl;

cin>>n;

int arr[n];

cout<<"enter the array elements"<<endl;

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

cin>>arr[i];

int key;

cout<<"enter the element to be searched"<<endl;

cin>>key;

linear\_search(arr,n,key);

}

}

**Output:**

enter the number of test cases

3

enter the number of elements

8

enter the array elements

34 35 65 31 25 89 64 30

enter the element to be searched

89

Present 6

enter the number of elements

5

enter the array elements

977 354 244 546 355

enter the element to be searched

244

Present 3

enter the number of elements

6

enter the array elements

23 64 13 67 43 56

enter the element to be searched

63

Not Present 6

Process returned 0 (0x0) execution time : 54.593 s

Press any key to continue.

**Question 3:** Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether a given key element is present in the sorted array or not. For an array arr[n], search at the indexes arr[0], arr[2], arr[4],.....,arr[2k ] and so on. Once the interval (arr[2k ] < key < arr[ 2k+1] ) is found, perform a linear search operation from the index 2k to find the element key. (Complexity < O(n), where n is the number of elements need to be scanned for searching)

#include<iostream>

#include<math.h>

using namespace std;

void Jump\_search(int arr[],int n,int key)

{

int comparisions=0,flag=0;

int step=sqrt(n),old=0;

while (arr[step-1]<key)

{

comparisions++;

old=step;

step+=sqrt(n);

if (step>n-1)

step=n;

}

for (int i=old;i<step;i++)

{

comparisions++;

if (arr[i]==key)

{

cout<<"Present"<<" "<<comparisions<<endl;

flag=1;

break;

}

}

if (flag==0)

cout<<"Not Present"<<" "<<comparisions-1<<endl;

}

int main()

{

int t;

cout<<"enter the number of test cases"<<endl;

cin>>t;

while (t--)

{

int n;

cout<<"enter the number of elements"<<endl;

cin>>n;

int arr[n];

cout<<"enter the array elements"<<endl;

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

cin>>arr[i];

int key;

cin>>key;

Jump\_search(arr,n,key);

}

}

**Output:**

enter the number of test cases

3

enter the number of elements

5

enter the array elements

12 23 36 39 41

41

Present 3

enter the number of elements

8

enter the array elements

21 39 40 45 51 54 68 72

69

Not Present 4

enter the number of elements

10

enter the array elements

101 246 438 561 796 896 899 4644 7999 8545

7999

Present 5

Process returned 0 (0x0) execution time : 74.453 s

Press any key to continue.