Program Linear Search

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

int main()

{

int t;

ifstream file;

file.open("input\_1.txt");

if(!file){

cout<<"File not found";

return 0;

}

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int num;

file>>num;

int comparison=0,flag=0;

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

comparison++;

if(num == arr[i]){

flag=1;

break;

}

}

if(flag==1)

cout<<"Present "<<comparison<<endl;

else

cout<<"Not Present "<<comparison<<endl;

}

file.close();

return 0

}

Program2 Binary Search

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

void Binary\_search(int\* arr, int l, int h,int key){

int mid,comparison=0,flag=0;

while(h>=l){

comparison++;

mid=(h+l)/2;

if(arr[mid]==key){

flag=1;break;

}else if(arr[mid]>key)

h=mid-1;

else if(arr[mid]<key)

l=mid+1;

}

if(flag)

cout<<"Present "<<comparison<<endl;

else

cout<<"Not Present "<<comparison<<endl;

}

int main()

{

int t;

ifstream file;

file.open("input\_2.txt");

if(!file){

cout<<"File not found";

return 0;

}

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int key;

file>>key;

Binary\_search(arr,0,n-1,key);

}

file.close();

return 0;

}

Program3 Exponential Search

#include <bits/stdc++.h>

#define endl "\n"

using namespace std;

void Exponential\_search(int\* arr, int n, int key){

int comparison=0,flag=0;

if(arr[0] == key){

comparison=1;

cout<<"Present "<<comparison<<endl;

exit(0);

}

int range=1;

while(range<n && arr[range]<=key){

comparison++;

range \*= 2;

}

int l=range/2,h=min(range,n-1),mid;

while(h>=l){

comparison++;

mid=(h+l)/2;

if(arr[mid]==key){

flag=1;break;

}else if(arr[mid]>key)

h=mid-1;

else if(arr[mid]<key)

l=mid+1;

}

if(flag)

cout<<"Present "<<comparison<<endl;

else

cout<<"Not Present "<<comparison<<endl;

}

int main()

{

ifstream file;

file.open("input\_3.txt");

if(!file){

cout<<"File not found!"<<endl;

return 0;

}

int t;

file>>t;

while(t--){

int n;

file>>n;

int arr[n];

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

file>>arr[i];

int key;

file>>key;

Exponential\_search(arr,n,key);

}

file.close();

return 0;

}