Page No. :
Date : / /

Name- Sushilkumar D. Dhamane

Class- SE I (F1) Date - 11/11/2021.

Sub - DSL (Data Stariure Lab)

Title - Search Algorithm.

Problem statement -

a) Write a python program to store Roll Nos of student in array who attended training program in random order. Write function for searching of a particular student attended using Linear search and sential search

b) Write a python program to store roll No.s. of student array who attended training program in sorted order Write a function for searchning if a particular student attended using Binary search and fibonacci search.

Objectives -

1) Implement Linear and sentineal search for randomly ordered array.

2) Implement Binary and Fibonacci search for a sorted array

Outcombes :- student will be able to

Write and execute python program that can perform Linear and sentinel search on unwrdered array and Binary and fibonacci search on ordered array.

Software and Hardware requirement -

window 10, Python 3.7, Pycharm, Intel - 9350 0HF CPU i5, 8 GB RAM, 512 SSD, 64-bit operating system, X64 - based processor.

Theory: Searchning is a techique of finding an element in given list standard search Algorithm are-

1) Linear sequential search: Examine elements starting from first and terminate process when list is exhausted or comparism result is a success.

2) sentinel search: An extra record equal to search element is replace by last element in list. search element is examine from starting and also with the replaced element and result is found.

3) Binary search: work on started array only comparism result is one of the following: i:= n/2.

a) key == arr[i] then return success.

b) key < arr[i] then call with left half

c) key > arr[i] the call with right half.

4) fibonacci search: works on sorted array only similar to Binary search but partition is not half, rather than fibonacii number.

$$f(n) = f(n-1) + f(n-2)$$

where $f(0) = 0$, $f(1) = 1$

Algorithm:

1) Algorithm for Linear search:

1) Start.

2) read list and read the element which need to be searched and index = 0

3) compare the searched element with each element in list if matched then write Index.

4) else write elemenent not found

5) stop.

2) Algorithm for sentinel searched.

1. Start.
2. Read the list which must be sorted and search element and i = 0
3. Replace the last element by searched element and compare the searched element with each element in list just as linear search,
4. If compare then reutrn i and if not compred then searched element with which is was replaced it mean with last element of list.
5. if it compared return index. else return element not foun
6. Stop.

3) Algorithm for Binary Searched:

1. Read the list and element to be searched.
2. Initialize the first term of element in list as zero and last element as len(list)-1)
3. find the mid as mid = (first + last)/2 and compare the mid with searched element if compared return m
4. if list[mid] is less than searched element change last (i.e upper bound) as last = mid+1 and repeat step3.
5. if change first (i.e lower found) as first = mid-1 and repeat step 3.

Algorithm for fibonacci:

1. n number of element present in list
2. A is the list of array which is sorted wher i=0
3. Find f(k) [where kth fibunacci Number.] which is greather than equal to n

Date : / /

4. if f(k) = 0 stop and write element not found
5. Initialize offset = -1.
6. S is the search element need to be Searched
7. i = min(offset + f(k-2), n-1)
8. if S is equal to A[i] then return I and stop the searched.
9. elif S > A[i] then k = k-1, offset = i and repeat 5 & 6
10. else S < [A[i] then k = k-2 repeat Step 5 & 6
11. Stop.

Pseucode.

1) Pseucode for linear search:

```
1. list = [ ]
2. x = int (input (''Enter the total No. of student'')
3. for i in range (x):
       y = int(input ( Enter Roll)
       i = i+1
       list.append(x)
   print(list)
4. for i in range(len(list)):
       if list[i] == key
       print(i)
       else:
         print(Not found).
```

Date : / /

2) Pseudocode for Sentinel:

```
1. list = [ ]
2. x = int (input (Enter total NO. of student)
3. for i in range (list):
        last = list [n-1]
        list [n-1] = key
        i = 0
    while (list [i] != key)
        i = i+1
    list [n-1] = last
    if ((i < n-1) or list [n-1] == key):
        return i
```

3. Pseudocode for Binary searched,

```
1. list = [ ]
2. first = 0  & last = len (list) - 1
3. While ( first <= last):
        mid = (first + last) // 2
    if ( list[mid] == key):
        return mid
    elif ( list[mid] < key):
        first = mid + 1
    else:
        last = mid - 1.
```

4. Pseudocode for Fibonacci search:

```
1. fib-series = [0,1]
2. fib1 = 0 & fib2 = 1
```

Date : / /

3. while(True):
    temp = ~~fibonacci~~fib2
    fib2 = fib1+fib1
    fib1 = Temp
    if (fib2 >= len(list)):
        fib-series.apped(fibond2)
        break
    else fib-series.append(fib2)
4. offset = -1
   while(fibm2-series[k] >1]:
       i = min(offset + fib-series[k-2], len(list)-1)
       if (key > list[i]):
           k = k-1
           offset = i
       elif (key < list[i]):
           k=k-2
       else:
           return i.
5. if (list[len(ist-1] = key):
       return len(list)-1
   else:
       return Not found.

Conclusion - Successfully implemented Search algorithm like linear & sentinel for random ordered & binary fibonacci for sorted array.

```
def sort(list,length):
    pre=1
    i=0
    while(i<length):
        while(True):
            x=int(input("Enter the Roll Numbers:"))
            break
        if (i!=0):
            if x not in list and x<pre:
                print("Enter the roll number greter than",pre)
            else:
                list.append(x)
                i=i+1
                pre=x
        else:
            list.append(x)
            i=i+1
            pre=x
    return list

def random(list,length):
    i=0
    while(i<length):
        while(True):
            try:
                x=int(input("Enter the Roll Numbers:"))
                break
            except valueError:
                print("Enter a valid input")
        if(x in list):
            print("Repeated rol number!")
        else:
            list.append(x)
            i=i+1
    return list

#Linear search
def linear(list,key): #time complexcity is Bigo(n)
    for i in range(len(list)):
        if(list[i]==key):
            return i
def result(index,key):
    if index is None:
        print("Student is not present in list!!")
    else:
        print("Roll number",key,"is present at index",index)
#Sentinel Search
#In Sentinel The element which need to be search is put at the last of the
list and then
#each element is compared just like linear search
def sentinal(list,n,key):    #n=number element in list
    last=list[n-1]
    list[n-1]=key
    i=0
    while(list[i]!=key):
        i=i+1
    list[n-1]=last
```

```
    if((i<n-1) or (list[n-1]==key)):  #i<n-1 whether we have search till last
or not
        return i
#Binary Search
def binary(list,key):   #Time complexcity Bigo(logn)
    first=0
    last=len(list)-1
    while(first<=last):
        mid=(first+last)//2
        if (list[mid]==key):
            return mid
        elif(list[mid]<key): #upper bound will change
            first=mid+1
        else:
            last=mid-1
    return False
def fibonacci(list,key):   #//Time complexcity Bigo(logn)
    fibonacci_series=[0,1]
    fibonacci1=0
    fibonacci2=1
    while(True):
        temp=fibonacci2
        fibonacci2=fibonacci2+fibonacci1
        fibonacci1=temp
        if(fibonacci2>=len(list)):
            fibonacci_series.append(fibonacci2)
            break
        else:
            fibonacci_series.append(fibonacci2)
        k=len(fibonacci_series)-1
        offset=-1
        while(fibonacci_series[k]>1):
            i=min(offset+fibonacci_series[k-2],len(list)-1)
            if(key>list[i]):
                k=k-1
                offset=i
            elif(key<list[i]):
                k=k-2
            else:
                return i
        if(list[len(list)-1]==key):
            return len(list)-1
        else:
            return None
A=int(input("Enter the total Number of student for random list:"))
B=[]
list1=random(B,A)
print()
C=int(input("Enter the total Number of student for Sorted list:"))
D=[]
list2=sort(D,C)
while(True):
    print("1.Random List\n2.Sorted List")
    choice=int(input("Enter the choice:"))
    print()
    if(choice==1):
        print("1.Linear Search\n2.Sentinal Search")
```

```
        choice=int(input("Enter the choice:"))
        print()
        if(choice==1):
            print(list1)
            x = int(input("Enter the Number to be searched:"))
            L=linear(list1, x)
            result(L, x)
            print()
        if(choice==2):
            print(list1)
            y=int(input("Enter the Number to be searched:"))
            S=sentinal(list1,A,y)
            result(S,y)
            print()
    if(choice==2):
        print("1.Binary Search\n2.Fibonacci Search")
        choice=int(input("Enter the choice:"))
        if(choice==1):
            print(list2)
            x=int(input("Enter the Number to be searched:"))
            B=binary(list2,x)
            result(B,x)
            print()
        if(choice==2):
            print(list2)
            y=int(input("Enter the Number to be searched:"))
            F=fibonacci(list2,y)
            result(F,y)
            print()
```