

Assignment 4

Class: SE IV

Roll No: 21430

Batch: F4

Dos: 16/09/2020

Problem Statement:

- Write a python program to store roll numbers of student in array who attended training program in random order. write function for searching whether particular student attended training program or not, using linear search and sentinel search.
- Write a python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using binary search and fibonacci search.

Learning objectives:-

- To write, learn and execute simple python program.
- To learn how to use function in python.
- To learn how to implement linear search, sentinel search, binary search and fibonacci search.

Learning outcomes:-

- Will be able to or aware of executing simple python code on various editors like pycharm, Eclipse etc.

- ii. Will be well aware of linear search, Sentinel search, binary search and fibonacci search and how to implement it.

Software Requirements:-

- i. Pycharm IDE community version 2020
- ii. 0.5 - Windows 10 64 bit

Hardware Requirements:-

- i. Windows 10
- ii. 4 GB Ram

Theory:-

- i. Linear Search:-

Linear search is the simplest searching algorithm that searches for an element in the list in a sequential order.

In linear search we start at one end and check every element until the desired element is not found.

Eg.

Consider list of five element and we want to search $K=3$ in it.

5	4	3	0	1
---	---	---	---	---

In linear search we start from 5 if first element is same as searching element K then it stop or it give continues searching as $K \neq 5$ and $K \neq 4$ when we come across second index $K=3$ so it will stop searching.

i. Sentinel Search:-

In sentinel search last element of array is replaced with the element to be searched and then the linear search is performed on the array.

The element to be searched will be definitely be found inside the array even if it was not present in the original array since the last element got replaced with it.

Eg. Consider array of 5 element on which we apply sentinel search.

5	4	2	0	1
---	---	---	---	---

And we want to search $k=3$ so, now last element is replace with 3.

The new array is

5	4	2	0	3
---	---	---	---	---

and now linear search is applied on above array to find number $k=3$.

iii. Binary Search:-

In binary search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of is less than middle array, narrow the interval to the lower half. otherwise narrow it to interval to the upper half.

Repeatedly check until the value is found or the interval is empty.

Eg. consider array of 5 elements and finding element $k=3$

2	3	6	8	9
---	---	---	---	---

firstly compare searching element with middle element if middle element is greater than k then we choose lower interval.

2	3	6
---	---	---

Now again we compare middle element of interval and our searching element is equal to middle element searching stops.

iv. Fibonacci search:-

Fibonacci search is method of searching a sorted array using divide and conquer algorithm.

Fibonacci search is a comparison based technique that uses fibonacci number to search an element in a sorted array.

* Algorithm:-

i) Linear Search:-

- step 1 : Set i to 1
- step 2 : if $i > n$ then go to step 7
- step 3 : if $A[i] = x$ then go to step 6
- step 4 : Set i to $i + 1$
- step 5 : Goto step 2
- step 6 : print element x found at index i and go to step 8.
- step 7 : print element not found
- step 8 : Exit.

ii) Sentinel Search:-

- step 1 : Replace last element of array with x
- step 2 : set i to 1
- step 3 : if $i > n$ then go to step 8
- step 4 : set i to $i + 1$
- step 5 : if $A[i] = x$ then go to step 7
- step 6 : Goto step 2
- step 7 : print element x found at index i and go to step 8
- step 8 : Exit.

iii) Binary Search:-

- Step 1 : search searching data from middle of array
- step 2 : if middle element $= x$ goto step 6
- Step 3 : else divide list using probing formula and find new middle.

Step 4 : IF x is greater than middle search in higher sublist.

Step 5 : if x is smaller than middle search in lower sublist.

Step 6 : if element = x print element found and go to step 7.

Step 7 : EXIT

iv) Fibonacci Search:-

The array of Fibonacci numbers is defined where $F_{k+2} = F_{k+1} + F_k$ when $k \geq 0$, $F_1 = 1$ & $F_0 = 0$.

Step 1 : Set $k = m$

Step 2 : IF $k = 0$, Step goto Step 7

Step 3 : compare item against element in F_{k-1}

Step 4 : if item matches go to Step 8

Step 5 : if item $< F_{k-1}$ change element from F_{k-1} to n , set $k = k - 1$ and go to Step 2.

Step 6 : if item $> F_{k-1}$ change element position from 1 to F_{k-1} , set $k = k - 2$ and go to Step 2.

Step 7 : print element not found & go to step 9

Step 8 : element found and go to step 9

Step 9 : EXIT

Test Cases:

No.	Description	Input	Expected o/p	Actual o/p	Status
1.	Searching element by linear search	$A = [2, 5, 7, 6, 4]$ $x = 5$	Found at index 1	found at index i	Pass
2	Searching element by Sentinel Search	$A = [2, 5, 7, 6, 4]$ $x = 1$	Found at index -1	found at index -1	pass

Conclusion:-

We learnt about linear search, binary search, sentinel search and fibonacci search and how to implement this search to find specific element


```
1  ls1=[]
2
3  def linear_search(x,ls2):
4      n = len(ls2)
5      for i in range(n+1):
6          if(x==ls2[i]):
7              print("Roll no is present at index ",i)
8              break
9      else:
10         print("Roll no is not found at any index")
11
12
13  def fibonacci_search(x,ls2):
14      n = len(ls2)
15      ls2.sort()
16      fib2 = 0
17      fib1 = 1
18      fib = fib2 + fib1
19      while (fib<n+1):
20          fib2 = fib1
21          fib1 = fib
22          fib = fib2 + fib1
```



```
23     offset = -1
24     while (fib>1):
25         i = min(offset + fib2, n)
26         if (ls2[i]<x):
27             fib = fib1
28             fib1 = fib2
29             fib2 = fib - fib1
30             offset = i
31         elif (ls2[i]>x):
32             fib = fib2
33             fib1 = fib1 - fib2
34             fib2 = fib - fib1
35         else:
36             return i
37     if (fib1 and ls1[offset + 1] == x):
38         return offset + 1
39     return -1
40
41
42 def binary_search(ls2, x):
43     min=0
44     n = len(ls2)
```

linear_search() > for i in range(n+1)

```
45 ls2.sort()
46 while min <= n:
47     mid = min + (n - min) // 2
48     if ls2[mid] == x:
49         return mid
50     elif ls2[mid] > x:
51         n = mid - 1
52     else:
53         min = mid + 1
54 return -1
```

```
56 def sentinel_search(x, ls2):
57     n = len(ls2)
58     last = ls2[n]
59     ls2[n] = x
60     i = 0
61     while (ls2[i] != x):
62         i += 1
63     ls2[n] = last
64
65     if ((i < n) or (x == ls2[n])):
66         return i
```

```
linear_search() > for i in range(n+1)
```



```

67     else:
68         return -1
69     n=int(input("How many Roll NUmber do you want to enter : "))
70     print()
71     for i in range(n):
72         m=int(input("Enter Roll Number : "))
73         ls1.append(m)
74     print(ls1)
75     while(1):
76         x=int(input("Enter roll no to search : "))
77         print("Enter 1 for linear")
78         print("Enter 2 for Sentinetal")
79         print("Enter 3 for Binary")
80         # print("Enter 4 for Fibonacci")
81         print("Enter 5 to EXIT")
82         a=int(input("Enter your choice : "))
83         if(a==1):
84             linear_search(x,ls1)
85         elif(a==2):
86             print("The index of roll no " +str(x) + " by sentinel search " +sentinel_search(x,ls1))
87         elif(a==3):
88             print("The index of roll no " +str(x) + " by Binary search " +binary_search(ls1,x))

```


linear_search() > for i in range(n+1)

```
print("Enter 3 for Binary")
# print("Enter 4 for Fibonacci")
print("Enter 5 to EXIT")
a=int(input("Enter your choice : "))
if(a==1):
    linear_search(x,ls1)
elif(a==2):
    print("The index of roll no ",x," by sentinel search ",sentinel_search(x,ls1))
elif(a==3):
    print("The index of roll no ",x," by Binary search ",binary_search(ls1,x))
elif(a==4):
    print("The index of roll no ", x, " by Fibonacci search ", fibonacci_search(x,ls1))
elif(a==5):
    break
else:
    print("-----Invalid Choice-----")
    print("Enter Choice Again")
```



```
How many Roll Number do you want to enter : >> 5
Enter Roll Number : >> 14
Enter Roll Number : >> 45
Enter Roll Number : >> 74
Enter Roll Number : >> 10
Enter Roll Number : >> 36
[14, 45, 74, 10, 36]
Enter roll no to search : >> 1
Enter 1 for linear
Enter 2 for Sentinetal
Enter 3 for Binary
Enter 5 to EXIT
Enter your choice : >> 1
Roll no is not found at any index
Enter roll no to search : >> 14
Enter 1 for linear
Enter 2 for Sentinetal
Enter 3 for Binary
Enter 5 to EXIT
>> |
```

> Special Variables

A vertical sidebar on the left side of the image contains several icons: a green play button, a red square, a green right-pointing triangle, a green gear, a grey gear, and a white plus sign.

```
Enter 2 for Sentinetal
Enter 3 for Binary
Enter 5 to EXIT
Enter your choice : >? 1
Roll no is present at index 0
Enter roll no to search : >? 1
Enter 1 for linear
Enter 2 for Sentinetal
Enter 3 for Binary
Enter 5 to EXIT
Enter your choice : >? 3
The index of roll no 1 by Binary search -1
Enter roll no to search : >? 36
Enter 1 for linear
Enter 2 for Sentinetal
Enter 3 for Binary
Enter 5 to EXIT
Enter your choice : >? 4
The index of roll no 36 by Fibbonacci search 2
Enter roll no to search :
>?
```