21118 Assignment No. 04 Name: Shubham Chemati Searching Operations Class: SE-1 (E-1) Page # Problem statement: aswerte a pythou program to store roll numbers of students in away who attended training program in Eardon order. Write function for searching wheather particular student attended trouwing program @ not unng b) Write a python program to store soll numbers of students In an array who attended training program in sated order write a function for searching who ther particular students attended training program @ not using binary searly & fibonaly search # Objective To lease of implement the following searching algorithms using python longuage 17 linear search ex Sentinel search 3> Brown search 4> filomaur Search. # ontrone learn has the searling algorithm work &
get knowledge of space & time complexity of that algorithms. Soffware Requirements: as: Windows 10 Python version: 3.8.5 Vs code (fex 6 editor): Nexsian 1.49.1 (Aug. 2020 version) # Hardware Regularement: manufactures: Acer Processor: Intel(R) 1.5 - 82650 CPU@ 1.60 GH2 Installed memory (RAM): 84B System Type. 64-bit 05, x 64-based architecture # Theory:

Classmate Date Rage

This a simplest searching algorithm:

In this algorithm, we check array element by element coquentially from first element to the last element.

If we get the match, stop checking array of the time the index we have found but it after the while while array we don't find element that means element that means element.

Alghough, it is simplest algorithm, the running time of algorithm is one because we have to their every element in worst case.

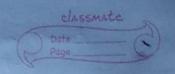
we don't need extra spare for the algorithm

27 September Search:

This algorithm is a modification of previous searthing algorithm is linear Search. In linear search there were some unneccessary compoursons which are avoided in sentine search.

In this algorithm the element to find (let x) is appended at the end of the dist we check hist element by element. After getting the element is check that index. It index is equal to last element that means element is not present in the list. Otherwise the element is present in the list. The time comparity of this algorithm is O(a) but due to less comparisons it is efficient than lived search.

This is most used searling algorithm in prairies would do to it's efficient time complexity.



This algorithm only works for socked acray in this algorithm we check a test middle dement of array if they matched we return middle index else if x is less than middle element we continue our search in left part else we continue our search in right part of array.

As in each iteration we eliminate half post of array (taking advantage of sorted lub) this algorithm eyns very fast. The Eynning time of algorithm is o(logn) which is very huge improvement over previous two searching algorithms.

4 Fibonacii Search

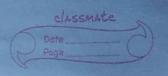
It is a method of searning a sorted array using duide & conquer algorithm that navous down possible location of element using generated fibonaui sequence. Some properties are of fibonaui sequence are used to reduce search area. Compared to binary search fibonaui search also gives same asymptotic time complexity ie. O(wlogn). The advantage of fibonaui search over binary search is that it doesn't uses complex bins operations like I multiplication & duision. It only und addition & substraction operation which gives this algorithm little edge

#- Algorithms (freudo Code)

Mingar Search

- 1. Algorithm linear Search (lub A, element 2): 1/2 eturns index of
- 2. for it to n. // n is sae of las
- 3 nif. A(1) = + 20: 1000 along 251100
- books etau i mass in man
- 5. Extrem 1

	Date
	rage
	ex Soutinel Seorch
	Consultation of the second of the second
1	Algorithm Southwel Search (Just, A, element x): / return index it is I A append (x) Resent XI
2	1 A append (x) Resent X
3	A append (x) The it is not the list of its list
4	. If ATi] = = x:
	break.
	if (iz=n): return i
	7. else return-1
1	
	37 Binary Search
	1. Algorithm Ringry Fearch (lup A, element x). 11 returns indent
	2. left to right to 1/4 is size of 1/p list.
	9. If (x == middle element of list)
	5. Zetuen tuder of middle element
	6. else if x is less than middle element)
	7. zight < mid-1 1) searching in left half.
111	4. else, it is greater than middle element)
V	1. left = mid+1 // searching 10 zight half.
1 200 10, US 1	10. Estaru -1. 11 H element not found return of
-	and principle addition to the plant of the state of the s
	9> fibonavi Search
	1. Algorithm fibo Search (lub A, element x); // returns index of 2. find small est fibonaus number greater x if present
70 mb	than @ equal to m. (let it be fm)
	Im+ , Im-1 -> nymbers preciding Im
	3. While (if there is an element in away):
	4. Compare & with range convexed by fm-2.
	also of a is loss or a late of the sless down
	else. If x is larger, more of the the step down
	J 770 770 2



8.	Untik for	last	remaining	Analo	el ousent
	- of car jo	Last	dernand	-	- George

class peclaration:

class search &

def _init_ (self):

Self. students = []

selfin = 0.

def. read Data (SUF):

11 rati reads data in self students

def. Sorb (self):

11 sorts self students.

def linear faich (setf)x):

11 linear search implement at on

def Sentine Scarch (Self)2)

11 Sentiarel search algorithm implementation

def. Brazy Search (xH, x):

let - Floonaui Search (setf, x)

11 Fibonaci seasch algorithm implementation

Analysis of Algorithms

Ī	Algorithm	Time Complexity	Space Complexity
,	Linear Search	searching the hist element	The algorithm doesn't
	Mark Themas	by element takes the	regular creation of any
		time proproctional to	additional data structure.
		number of element in lab.	If works on original
		. In aymptotic notation,	list hence in
		time complexity of linear	asymptotic notation it
		search is oca)	require cont. space.

	0/4-	
1	Date	464
	Date	200
	Page	
12	-	-

			Date Page
,	Soutine) Search	this algorithm also grath the list element by element some as trucar search hence gives asymptotic time complainty as o(u)	Scarch algorithm doesn't require any additional space is a constant algorithm
3.	Pinary Search.	THE CONTRACTOR OF THE PARTY OF	Maux any additions space, hearce it is constant space algorithm.
	and the second	a_{b} $h=1$, $T(1)=c$. $h=1$ $\Rightarrow n=2^{k}$ 2^{k} $k=\log n$ J_{2}	on militario
100 ho	matissile est authorism since	- M(n) = kc + T(1) = kc + c = logn (c+1) = 0 (logn)	A for sixting to A made and a mad
	AND REPORT OF THE PARTY OF THE	the asymptotic time couple thy of binary search is logarithmic re o (log n)	

9>	Fibonau Search	Same as binary saech If we consider no
		this algorithm eliminate extra memory for
		N/s @ ~2/e past of fibonaci sequence the
		list in each iteration algorithm is constant
	DELAS SAN SANS	hence gives logatithmic space algorithm.
		time complexity
		Asymptotic time complexity
		of algorithm is
		of algorithm is O(logn)
	and from and the	The State of the S

Test Cases:

Test cours.	Input given	expected ontput	Actual output.
1.			researched =
29533	[2,3,5,1] for x=3	Index: 1	index: I
	of anationals and the	home bearing	mosoo ÷
27	Linear search on	Nob present	index: -1
	[2,3, 6,4,1] for x=5		
	, , , , , , , , , , , , , , , , , , , ,	The Raw San Electric	
3)	Sentine Search on	Index: 1	rnden:
	[2,3,5,1] for x=3		THE PARTY OF THE P
3>	Sentine fearch on	Not present	Indea: -1
	[2,3,6,4,1] for x=5		
	[2/3/8/11]	THE THE PARTY	
4)	Para Canala Dia	Index:3	index:3
	Brnazy search on (1) (1) (1)		
	For x=7.	A STATE OF	
	100 70- 4		
4	Binary search on [1/3,5,7,9]	Indo b present	index: 1
-/		(
	For x = 6		

			Classmate Page
	Fibonaui Search on [1,3,7,1],15) For	Nob present	Index:4
7)	7=12. Fibonai Search on	inden = J	Index:3
	2=11 for	A STATE OF THE STA	
#	Applications The searching algori	thms are one of t	the most used
	algorithms in compute ap	plication in every	field of induly
#	Conclusion:	the a wood ment I'	m able to use
	+ program general various applications	seasolving autisign a	gosithms for.
	L MARKET LAN	da ex angle	2 Sutual Cas
	Tanking Jane	- Inter-	A Section Leave
		alai sa	Jan 2003 6
		# F 7	E 1170 ·
	- Palser Janana	old The Francisco	der to the design of the second