20DCS103

Assignment : 01

DSA

Que: 1 : For given float array [-12 : -4 , -5 : 5 , 14 : 25] with base address 5006, find the address of [-8, 5, 19] if storage in row and column major order.

→ Address of A[i, j, k] is given by

$$Base + size[(D - D_0)RC + (i - R_0)C + (j - C_0)]$$

which is in row major order

→ Address of A[i, j, k] is given by

$$Base + size[(D - D_0)RC + (i - R_0) + (j - C_0)R]$$

which is in column major order

∴ Here, $i = -8$, $j = 5$, $k = 19$

D = total number of cells depth wise
$= (25 - 14) + 1$
$= 12$

$D_0 = 14$

R = total number of rows
$= -4 - (-12) + 1$
$= -4 + 12 + 1$
$= 9$

ID : 20DCS103

4DCSE2 : B

Date
Page

$R_0 = -12.$

c = total number of columns
$= 5 - (-5) + 1$
$= 11$

$C_0 = -5$

Base address is 5006 and for float array size of each element is 4 bytes

∴ Address using row major order

$$= 5006 + 4[(12-14)(9)(11) + (-8+12)(11) + (5+5)]$$
$$= 5006 + 4[-198 + 44 + 10]$$
$$= 5006 + 4(-144)$$
$$= 5006 - 576$$
$$= \boxed{\cancel{4430}}$$

and address using column major order.

$$= 5006 + 4[(12-14)(9)(11) + (-8+12) + (5+5)9]$$
$$= 5006 + 4[-198 + 4 + 90]$$
$$= 5006 + 4(-104)$$
$$= 5006 - 416$$
$$= \boxed{4590.}$$

Scanned with CamScanner

Date
Page

Que: 2 Arrange the given data in ascending order using bubble sort, selection sort and insertion sort. Show all steps.
3, 23, 32, -32, -3.

* Bubble sort :- n = sizeof(arr) / sizeof(arr[0])

```
for i ← 0 up to n
    for j ← 0 up to n - i - 1
        if arr[j] > arr[j+1]
            exchange( arr[j], arr[j+1])
```

1st pass : i j 3, 23, 32, -32, -3.

	i	j		
1)	0	0	→	3, 23, 32, -32, -3
2)	0	1	→	3, 23, 32, -32, -3
3)	0	2	→	3, 23, -32, 32, -3
4)	0	3	→	3, 23, -32, -3, $\boxed{32}$ ← sorted

2nd pass : i j 3, 23, -32, -3, 32

	i	j		
1)	1	0	→	3, 23, -32, -3, 32
2)	1	1	→	3, -32, 23, -3, 32
3)	1	2	→	3, -32, -3, $\boxed{23, 32}$

↑ sorted

3rd pass : i j 3, -32, -3, 23, 32

	i	j	
1)	2	0	-32, 3, -3, 23, 32
2)	2	1	-32, -3, $\boxed{3, 23, 32}$
~~3)~~	~~2~~	~~2~~	

↑ sorted

4th pass : i j -32 , -3 , 3 , 23 , 32

1) 3 0 → $\boxed{-32}$ | -3 , 3 , 23 , 32 |
sorted

∴ After sorting : -32 , -3 , 3 , 23 , 32.

** Insertion sort :-

for i ← 1 up to n
key ← arr[i]
p ← i-1

while p >= 0 and arr[p] > key
arr[p+1] ← arr[p]
~~p ← p - 1~~
~~arr[p+1]~~ ← key

1st pass i p key 3 | 23 , 32 , -32 , -3.
sorted

1) 1 0 23 → 3 , 23 , | 32 , -32 , -3.
sorted

2nd pass i p key 3 , 23 , 32 , -32 , -3

1) 2 1 32 → 3 , 23 , 32 , -32 , -3

2) 2 0 32 → 3 , 23 , 32 | -32 , -3
sorted

Date
Page

3^{rd} pass i j key 3, 23, 32, -32, -3

1) 3 2 -32 → 3, 23, -32, 32, -3
2) 3 1 -32 → 3, -32, 23, 32, -3
3) 3 0 -32 → -32, 3, 23, 32 | -3
sorted

4^{th} pass i j key -32, 3, 23, 32, -3.

1) 4 3 -3 → -32, 3, 23, -3, 32
2) 4 2 -3 → -32, 3, -3, 23, 32
3) 4 1 -3 → -32, -3, 3, 23, 32
4) 4 0 -3 → -32, -3, 3, 23, 32
sorted

∴ After sorting : -32, -3, 3, 23, 32.

** Selection sort :- 3, 23, 32, -32, -3

for i ← 0 up to n-1
  for j = k ← i up to n
    if arr[j] < arr[k]
      then k ← j
  exchange (arr[i], arr[k])

1^{st} pass : i j k   k → 3, 23, 32, -32, -3.

1) 0 1 0   3, 23, 32, -32, -3
2) 0 2 0   3, 23, 32, -32, -3.
3) 0 3 0   3, 23, 32, -32, -3

Date
Page

4) 0 4 3 → $\boxed{3}$, 23, 32, $\boxed{-32}$, -3.
swap

∴ After 1st pass : -32, 23, 32, 3, -3.

2nd pass.
-32 | 23, 32, 3, -3.
sorted

-32, 23, 32, 3, -3.
i, k, j

1) -32, 23, 32, 3, -3.
i,k, j

2) -32, 23, 32, 3, -3.
i, k, j

3) -32, $\boxed{23}$, 32, 3, $\boxed{-3}$.
swap

4) ~~-32, -3 | 32, 3, 23~~
sorted

3rd pass
-32, -3, 32, 3, 23.
i,k, j

1) -32, -3, 32, 3, 23.
i, k, j

2) -32, -3, $\boxed{32}$, $\boxed{3}$, 23.
k, j
swap

3) -32, -3, 3 | 32, 23
sorted

Scanned with CamScanner

4^{th} pass :- -32, -3, 3, 32, 23
i,k ↑ j ↑

1) -32, -3, 3, 32, 23
i ↑ k ↑

swap

2) -32, -3, 3, 23, 32.

∴ After sorting : -32, -3, 3, 23, 32

Que:- 03 Convert given infix notation into it's equivalent reverse polisnotation using stack.

post fix

A + (B * P - (D / E - F) * G) * K

Expression	Current Symbol	Action taken	Stack	Output
A	—	—	—	—
+	A	print A		A
(	+	push +	+	A
B	(	push (	(+	A
*	B	print B	(+	AB.
P	*	push *	*(+	AB.
—	P	print P	*(+	ABP
~~(~~	~~-~~	~~push -~~	~~-*(+~~	~~ABP~~
~~D~~	~~(~~	~~push (~~	~~(-*(+~~	~~ABP~~
~~/~~	~~D~~	~~print D~~	~~(-*(+~~	~~ABPD~~

Date
Page

Expression	Current Symbol	Action taken	Stack	Output
C	-	pop * , push -	-(+	ABP*
D	C	push C	C-(+	ABP*
/	D	print D	C-(+	ABP*D
E	/	push /	/C-(+	ABP*D
-	E	print E	/C-(+	ABP*DE
F	-	pop / , push -	-(-(+	ABP*DE/
)	F	print F	-(-(+	ABP*DE/F
*	)	pop -	-(+	ABP*DE/F-
G	*	push *	*-(+	ABP*DE/F-
)	G	print G	*-(+	ABP*DE/F-G
*	)	pop *, -	+	ABP*DE/F-G*-
K	*	push *	*+	ABP*DE/F-G*-
	K	print K	*+	ABP*DE/F-G*-K

∴ Result : ABP* DE/F -G*-K *+

Que: 04 Construct a Binary Search Tree.

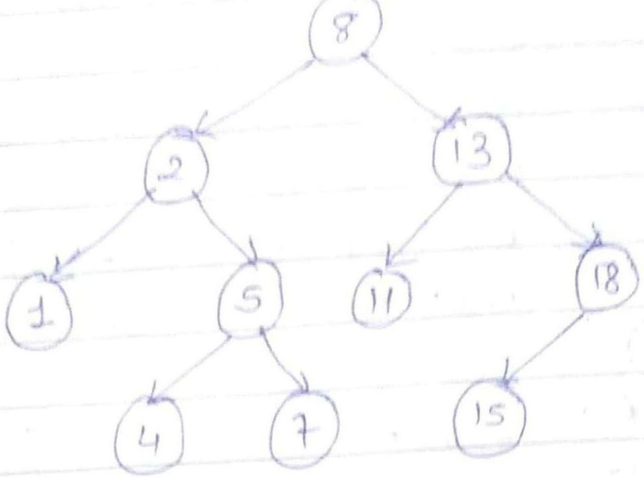
→ Binary search tree is a node based binary data structure which has following properties :

1) The left subtree of a node contains only nodes with keys lesser than the node's key.

2) The right subtree of a node contains only nodes with keys greater than the node's key.

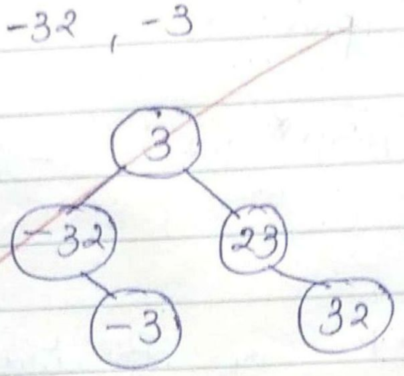
3) The left and right subtree each must also be a binary search tree. There must be no duplicate node.

Ex:-



→ Now constructing bst of given sequence.

3, 23, 32, -32, -3



Que: 05 For the given hash function, calculate the index, store data and resolve collision using linear probing. Size of array is 8.

$H(x) = (x \% 6) + 2$.

11, 111, 11, 10, 1

→ Here, $h(x) = (x \% 6) + 2$.

and 11, 111, 11, 10, 1 are given

→ $x = 11$ $\therefore$ $h(11) = (11 \% 6) + 2$

$= 5 + 2 = 7$.

$\therefore$ 11 will be stored at index 7.

0	11
1	
2	1
3	
4	
5	111
6	10
7	11

→ $x = 111$

$\therefore h(111) = (111 \% 6) + 2$

$= 3 + 2 = 5$.

$\therefore$ 111 will be stored at index 5.

→ $x = 11$

$\therefore h(11) = (11 \% 6) + 2$

$= 5 + 2 = 7$.

$\therefore$ 11 will be stored at index 7 but 11 is already there which results in collision.

$\therefore$ 11 will be stored at the next empty location which comes after index 7. if there is no any empty space, then we should start from the beginning.

$\therefore$ 11 will be stored at index 0.

→ $x = 10$

$\therefore h(10) = (10 \% 6) + 2$

$= 4 + 2 = 6$.

$\therefore$ 10 will be stored at location 6.

→ $x = 1$

$\therefore h(1) = (1 \% 6) + 2$

$= 1 + 2 = 3$.

∴ 1 will be stored at index 3.

→ After all the array we get is.

11		1			111	10	11
0	1	2	3	4	5	6	7

——— X ———

12/4/22