Synoptic for MSE MCA
Data structures. 14/3/2018.

Q-1) Given data,

10, 20, 2, 3, 15, -22.

Bubble Sort:-

Pass \ Index	0	1	2	3	4	5
0	10	20	2	3	15	-22
1	10	~~20~~ 2	~~20~~ 3	~~20~~ 15	~~20~~ -22	20
2	~~10~~ 2	~~10~~ 3	10	~~15~~ -22	15	20
3	2	3	~~10~~ -22	10	15	20
4	2	~~3~~ -22	3	10	15	20
5	-22	2	3	10	15	20.

(2 mks)

~~Select~~
Insertion Sort:-

Pass \ Index	0	1	2	3	4	5
0	10	20	2	3	15	-22
1	10	20				
2	2	10	20			
3	2	3	10	20		
4	2	3	10	15	20	
5	-22	2	3	10	15	20.

(2 mks)

Bubble sort:-

No. of passes:- 5
No. of Iterations:- 25 {2 mks}

Insertion sort:-

No. of passes:- 5
No. of Iterations:- (1+2+1+4+1) = 9 {2 mks}

Q-2) Double Hashing:-

Size = 11
Array = 15, 17, 88, 59.

Formula for double Hashing,

$$h(k,i) = (h_1(k) + i h_2(k)) \bmod m$$

where,

$$h_1(k) = h(k) = k \bmod m$$

$$h_2(k) = 1 + (k \bmod (m-1))$$

88	0
	1
	2
	3
15	4
	5
17	6
	7
	8
	9
59	10.

(i) For (15)

$$h(15,0) = (h_1(15) + 0\,h_2(15)) \bmod 11$$

where,

$$h_1(15) = 15 \bmod 11 = 4$$

$$\therefore h(15,0) = (4+0) \bmod 11$$
$$= 4.$$

Hence, Data 15 will be stored at location 4.

(ii) For (17)

$$h(17,0) = (h_1(17) + 0\,h_2(17)) \bmod 11$$

where,

$$h_1(17) = 17 \bmod 11 = 6.$$

$$\therefore h(17,0) = (6+0) \bmod 11$$
$$= 6.$$

Hence, 17 will be stored at location 6.

(iii) For 88,

$$h(88,0) = (h_1(88) + 0\,h_2(88)) \bmod 11$$

where,

$$h_1(88) = 88 \bmod 11 = 0.$$

$$\therefore h(88,0) = (0+0) \bmod 11$$
$$= 0.$$

Hence, 88 will be stored at 0.

(iv) For 59.

$h(59,0) = (h_1(59) + 0\, h_2(59)) \bmod 11$

where,

$h_1(59) = 59 \bmod 11 = 4$

$\therefore h(59,0) = (4+0) \bmod 11$

$= 4$

There is a collision, so 59 can't be stored at location 4.

Hence, calculate $h(59,1)$

$h(59,1) = (h_1(59) + 1 \cdot h_2(59)) \bmod 11$

where

$h_1(59) = 4$

$h_2(59) = 1 + (59 \bmod 10)$

$= 1 + 9$

$= 10.$

Hence, 59 will be stored at location 10.

For each of data mapping (2 mks) have been allotted.

Q-3) Interpolation search.

Key element = 30.

a[8] = { 10, 20, 30, 40, 50, 60, 70, 80 }.

Ans:- (i) For applying Interpolation search, data should be sorted. Here in this, case, data is already sorted. (1 mk)

(ii) Calculate values of low, high & mid.

here, low = 0 (3 mks)

high = 7

mid = low + (high - low) * ((key - a[low]) / a[high] - a[low]))

	low							high
Index	0	1	2	3	4	5	6	7
value	10	20	30	40	50	60	70	80.

$\therefore$ mid = 0 + (7 - 0) * ((30 - 10) / (80 - 10))

= 0 + 7 * (20/70)

$\therefore$ mid = 2

(iii) Compare a[mid] with key value, if it is equal search is successful, else adjust low & high pointer. and perform interpolation search again. (1 mk)

Here, in this case, a[mid] = 30 = key.

Hence, key 30 found at location 2.

Q-3) Binary search
key = 35

a[5] = {2, 44, 35, 88, 1}.

Ans:- (i) For binary search, the array should be sorted.
The sorted array is, {1 mk}

a[5] = {1, 2, 35, 44, 88}. {By applying bubble sort}.

(ii) Calculate values of low, high & mid

low = 0

high = 4

mid = (low + high)/2 {3 mks}

= (0 + 4)/2 = 2

Index	0 (low)	1	2 (mid)	3	4 (high)
value	1	2	35	44	88

(iii) Compare a[mid] with key value. If they are same then search is successful else adjust low & high pointer & perform binary search again.

Here, a[mid] = 35 = key. {1 mk}

Hence, 35 is found at location 2.

Q-4) $T(n) = 2T(n/2) + n$

→ (i) Compare this with following eqn.

$T(n) = aT(n/b) + f(n)$. (1mk)

Hence, $a=2$, $b=2$ & $f(n)=n$.

(ii) Evaluate $n^{\log_b a}$

here, $n^{\log_2 2} = n^1 = n$. (1 mk)

(iii) If $n^{\log_b a} > f(n)$ Then $O(n) = n^{\log_b a}$

Else if $n^{\log_b a} < f(n)$ Then $O(n) = f(n)$

else $O(n) = f(n) \log n$.

(2 mks)

Here in this case,

$n^{\log_b a} = f(n)$

Hence, The worst case complexity is

$O(n) = n \log n$.

Q-4) Radix sort.

a[5] = {1234, 234, 4564, 898, 9890}.

Ans:-

(i) a[5] = {1234, 0234, 4564, 0898, 9890}. (1mk)

Append 0's on L.H.S. if the no. of digits are less than 4 digits.

(ii) Pass 1:- {consider L.S.D. of a no.}. (1mk)

0	1	2	3	4	5	6	7	8	9
9890				1234				0898	
				0234					
				4564					

After pass 1, elements are,

9890, 1234, 0234, 4564, 0898.

(iii) Pass 2:- {consider second L.S.D. of a no.} (1mk)

0	1	2	3	4	5	6	7	8	9
			1234			4564			9890
			0234						0898

After pass 2, elements are,

1234, 0234, 4564, 9890, 0898.

(iv) Pass 3:- {consider 3rd L.S.D. of a no.}. (1mk)

0	1	2	3	4	5	6	7	8	9
		1234			4564			9890	
		0234						0898	

Hence, After pass 3, Content of array are,

1234, 0234, 4564, 9890, 0898.

Q-5) Sparse Matrix -

$$M = \begin{array}{c|c} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 2 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \end{array}$$

→ Here,

The Triplet representation of above M is,

Row	column	value.
3	3	3
0	1	2
1	0	3
2	2	3.

(1mk)

For linked list representation, we use 2 nodes which are as follows,

(1mk)

Header Node

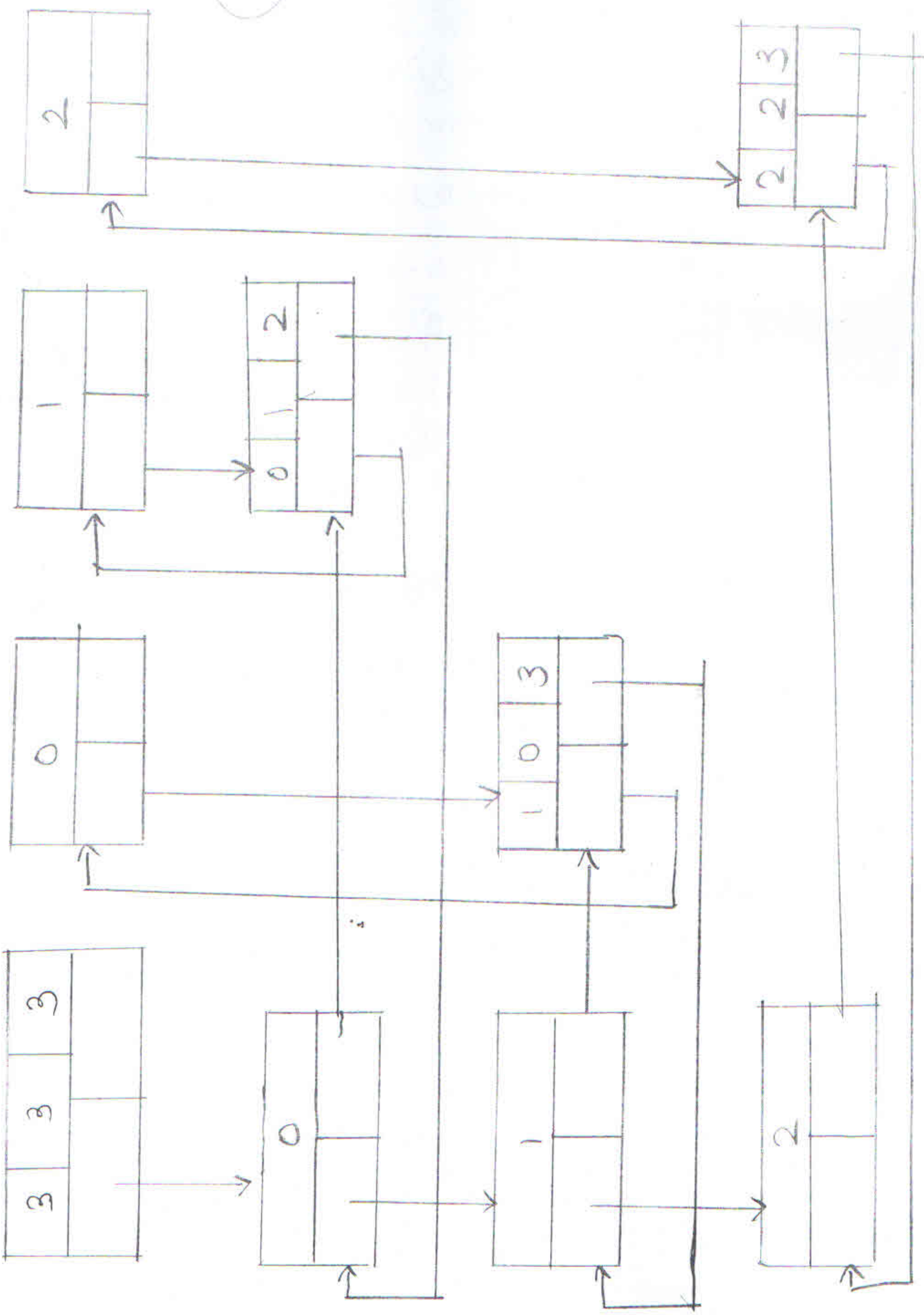
Index value	
down	Right

Element Node.

row	column	value
up/down	left/right	

The linked list representation of above M is,

(3mks)



Q-5) Algorithm for searching an element in the Singly linked list

Consider list pointer points to very first node.

Begin

Take an element to be searched from user, in data1

Initialize q with list

Initialize count with 0.

Repeat

Count is incremented by 1

Until q→data != data1 and q→next != NULL

If q→data = data1

Print element data1 found at location count

Else

Print element not found.

End.