Name:- S.Neda Anjum
Regno: 192311071
Subcode:- CSA0672

(20) Give an array of [4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8, 11,-9] integers, sort the following elements using insertion sort using Brute force Approach strategy analyse complexity of the algorithm.

Step1:-
4 -2 5 3 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step2:-
-2 4 5 3 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step3:-
-2 ~~3~~ 4 3 5 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step4:-
-2 3 4 5 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step5:-
-2 3 4 5 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step6:-
-2 3 4 5 -5 10 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step7:-
-5 -2 3 4 5 10 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step8:-
-5 -2 3 4 5 2 10 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step9:-
-5 -2 2 3 4 5 10 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step10:-
-5 -2 2 3 4 5 8 10 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step11:-
-5 -2 2 3 4 5 -3 10 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step12:-
-5 -3 -2 2 3 4 5 10 6 7 -4 1 9 -1 0 -6 -8 11 -9

Step 13:-

-5 -4 -3 -1. 1 2 3 4 5 6 7 8 9 10 0 -6 -8

Step14:-

-5 -4 -3 -1 1 0 2 3 4 5 6 7 8 9 10 -6 -8 11 -9

Step15:-

-5 -4 -3 -1 0 1 2 3 4 5 6 7 8 9 -6 10 -8 11 -9

Step16:-

-6 -5 -4 -3 -1 0 1 2 3 4 5 6 7 8 9 10 -8 11 -9

step17:-

-8 -6 -5 -4 -3 -1 0 1 2 3 4 5 6 7 8 9 10 11 -9

step18:-

-9 -8 -6 -5 -4 -3 -1 0 1 2 3 4 5 6 7 8 9 10 11.

Insertion sort time complexity:-

Best case:- O(n), if the list is already sorted, where n is no of elements in the list

Average case:- $O(n^2)$, if the list is randomly ordered.

worst case:- $O(n^2)$, if the list is in reverse order.

19) Sort the following elements using insertion sort using Brute force Approach strategy [38,27,43,3,9,82,10,15,88, 52,60,5] and analyse complexity of the algorithm.

Given

[38,27,43,3,9,82,10,15,88,52,60,5]

1:- 38 27 43 3 9 82 10 15 88 52 60 5

i ⟶ j

2:- 27 38 43 3 9 82 10 15 88 52 60 5

i j

3:- 27 38 43 3 9 82 10 15 88 52 60 5

i j

4:- 3 27 38 43 9 82 10 15 88 52 60 5

i j

5: 3 27 38 43 9 82 10 15 88 52 60 5
i j

6:- 3 9 27 38 43 82 10 15 88 52 60 5
i j

7:- 3 9 27 38 43 82 10 15 88 52 60 5
i j

8:- 3 9 10 27 38 43 82 15 88 52 60 5
i j

9:- 3 9 10 15 27 38 43 82 88 52 60 5
i j

10:- 3 9 10 15 27 38 43 52 82 88 60 5
i j

11:- 3 9 10 15 27 38 43 52 60 82 88 5
i j

12:- 3 9 10 15 27 38 43 52 60 82 5 88
i j

13:- 3 9 10 15 27 38 43 52 60 5 82 88
i j

14:- 3 5 9 10 15 27 38 43 52 60 82 88

Insertion sort time complexity:

Best case:- $\Theta(n)$ if the list is already sorted, where n is number of elements in the list.

Avg case:- $\Theta(n^2)$ if the list is randomly ordered.

Worst case:- $\Theta(n^2)$ if the list is in reverse order.

8) Sort the array 64, 25, 12, 22, 11 using selection sort. What is the time complexity of selection sort in the best, worst and average cases?

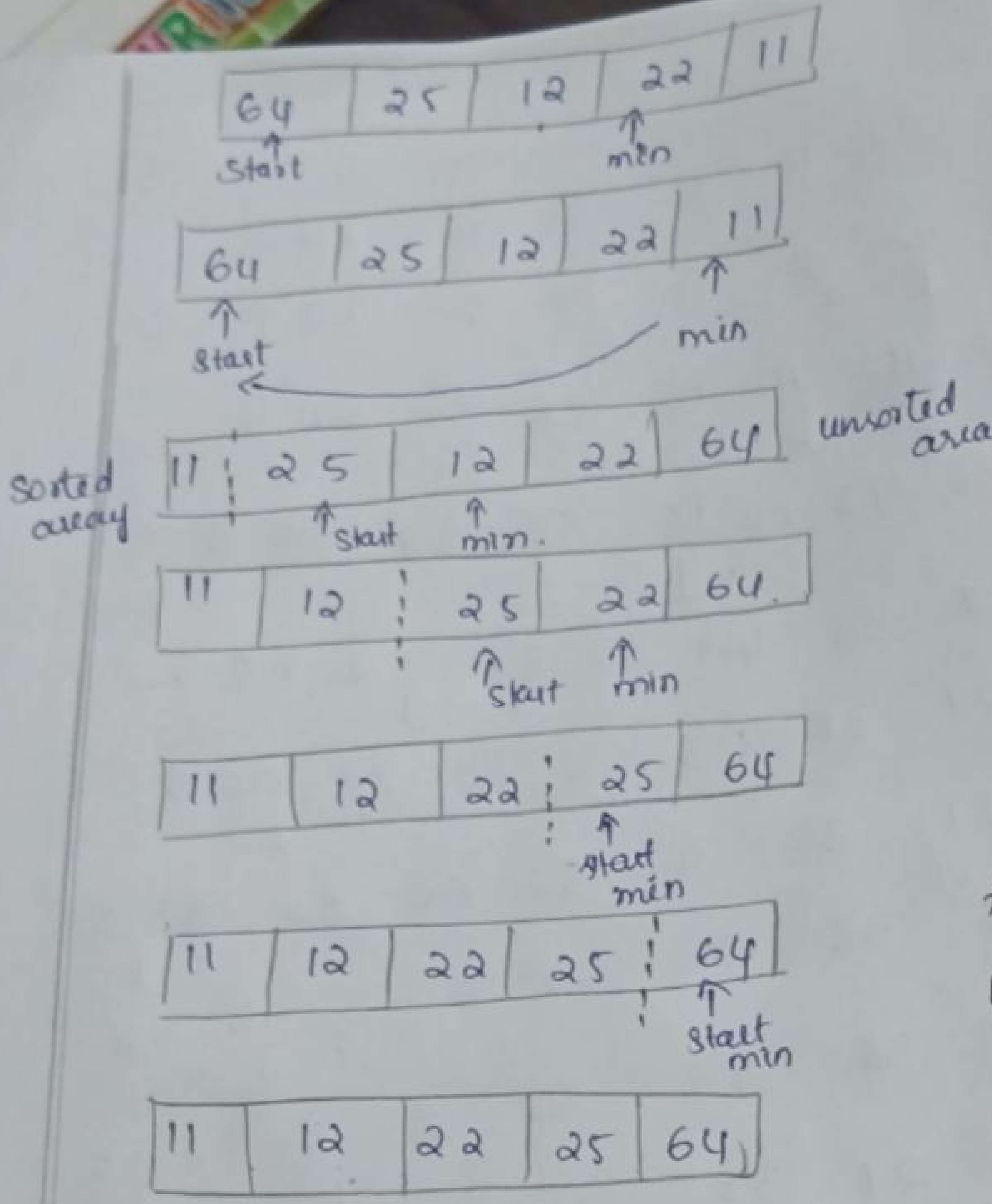
Given

64	25	12	22	11

↑ start ↑ min

64	25	12	22	11

↑ start ↑ min.



Time complexity

Best case:- $O(n^2)$

Avg case :- $O(n^2)$

worst case:- $O(n^2)$

17) Sort the array 64, 34, 25, 12, 22, 11, 90 using Bubble s
What is the time complexity of selection sort i
Bubble
the best, worst, & average cases?

Given

64 34 25 12 22 11 90
I → ↑p ↑j

34 64 25 12 22 11 90
↑i j

34 25 64 12 22 11 90
i j

34 25 12 64 22 11 90
i j

34 25 12 22 64 11 90
i j

34 25 12 22 11 64 90
i j

```
ii →  34  25  12 · 22  11  64  90
      i   j
      25  34  12  22  11  64  90
          i   j
      25  12  34  22  11  64  90
              i   j
      25  12  22  34  11  64  90
                  i   j
      25  12  22  11  34  64  90
                      i   j
      25  12  22  11  34  64  90
                          i   j

iii → 25  12  22  11  34  64  90
      i   j
      12  25  22  11  34  64  90
          i   j
      12  22  25  11  34  64  90
              i   j
      12  22  11  25  34  64  90
                  i   j
      12  22  11  25  34  64  90
                          i   j

iv    12  22  11  25  34  64  90
      i   j
      12  22  11  25  34  64  90
          i   j
      12  11  22  25  34  64  90
              i   j
      12  11  22  25  34  64  90
                      i   j
      12  11  22  25  34  64  90
                          i   j

      12  11  22  25  34  64  90.
```

24 34, 64, 90

IV) 12 11 22 24 34 64 90
i j

11 12 22 24 34 64 90
i j

11 12 22 24 34 64 90
i j

11 12 22 24 34 64 90
i j

11 12 22 24 34 64 90
i j

11	12	22	24	34	64	90

Time complexity:

Best case - O(n)

worst case - $O(n^2)$

Avg case - $O(n^2)$

16) Sort the following elements using merge sort divide-and-conquer strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] & analyse complexity of the algorithm.

Given $\overset{0}{38}$ $\overset{1}{27}$ $\overset{2}{43}$ $\overset{3}{3}$ $\overset{4}{9}$ $\overset{5}{82}$ $\overset{6}{10}$ $\overset{7}{15}$ $\overset{8}{88}$ $\overset{9}{52}$ $\overset{10}{60}$ $\overset{11}{5}$

$M = \frac{l+h}{2} = \frac{0+11}{2} = \frac{11}{2} = 5.5 \simeq 6$

$\overset{0}{38}$ $\overset{1}{27}$ $\overset{2}{43}$ $\overset{3}{3}$ $\overset{4}{9}$ $\overset{5}{82}$ $\overset{6}{10}$ | $\overset{7}{15}$ $\overset{8}{88}$ $\overset{9}{52}$ $\overset{10}{60}$ $\overset{11}{5}$

$m = \frac{l+h}{2} = \frac{0+6}{2} = 3$ $m = \frac{l+h}{2} = \frac{7+11}{2} = \frac{18}{2} = 9$

$\overset{0}{38}$ $\overset{1}{27}$ $\overset{2}{43}$ $\overset{3}{3}$ | $\overset{4}{9}$ $\overset{5}{82}$ $\overset{6}{10}$ | $\overset{7}{15}$ $\overset{8}{88}$ $\overset{9}{52}$ | $\overset{10}{60}$ $\overset{11}{5}$

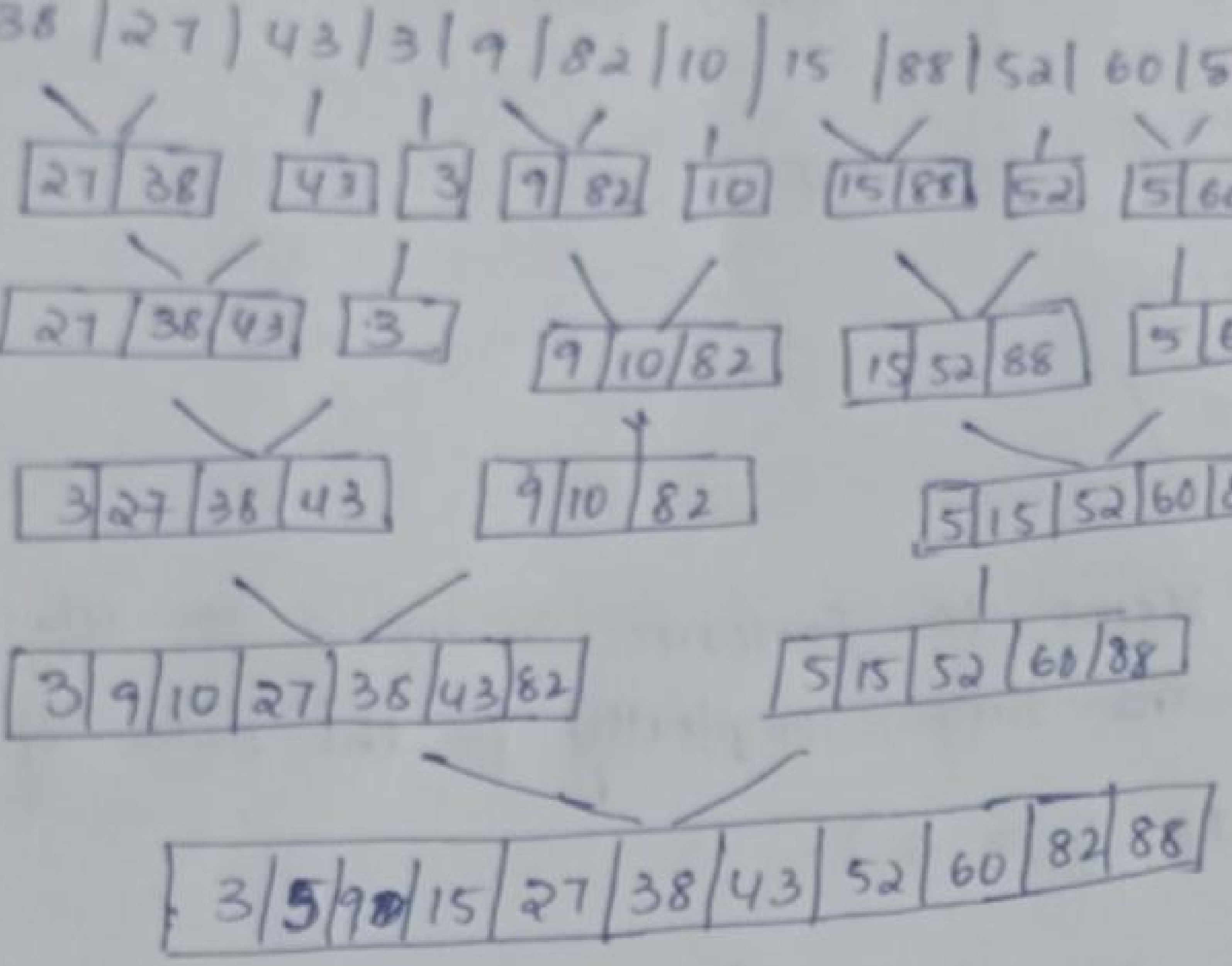
$m = \frac{l+h}{2} = \frac{0+3}{2} = 1.5 \simeq 2$ $m = \frac{l+h}{2} = \frac{10}{2} = 5$ $m = \frac{7+9}{2} = 8$ $m = \frac{l+h}{2} = 0$

$\overset{0}{38}$ $\overset{1}{27}$ $\overset{2}{43}$ | $\overset{3}{3}$ | $\overset{4}{9}$ $\overset{5}{82}$ | 10 | $\overset{7}{15}$ $\overset{8}{88}$ | 52 | 60 | 5

$m = \frac{0+2}{2} = 1$

$\overset{0}{38}, \overset{1}{27}$ | 43 | 3 | 9 | 82 | 10 | 15 | 88 | 52 | 60 | 5

m = 0



Time complexity

~~Time~~ Best case - $O(n^2)$
Avg case - $O(n^2)$
Worst case - $O(n^2)$

) Find the index of the target value 10 using binary search from the following list of elements [2, $\overset{4,}{}$6, 8, 10, 12, 14, 16, 18, 20]

0	1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18	20

$B = \frac{l+h}{2} = \frac{0+9}{2} = \frac{9}{2} = 4.5 \simeq 4$

0	1	2	3	4	5		6	7	8	9
2	4	6	8	10	12	|	14	16	18	20
		P_1						P_2		

Key = 10

l	h	mid	condition
0	9	4	A[mid]==key return result

$B = \frac{l+h}{2} = \frac{0+9}{2} = 4.5 \simeq 4$

key value = 10

$\boxed{\text{Index } 4 = 10}$

14) Find the no. of items to perform swapping for selection sort. Also estimate the time complexity for the order of notatio

set S(12, 7, 5, -2, 18, 6, 13, 4).

Given, S{12, 7, 5, -2, 18, 6, 13, 4)

12 7 5 -2 18 6 13 4

↑ ↑ ↑ ↑

start min min min

-2 ⁞ 7, 5, 12, 18, 6, 13, 4

↑ ↑ ↑ ↑

start min min min

-2, 4 ⁞ 5, 12, 18, 6, 13, 7

↑

start

min

-2, 4, 5 ⁞ 12, 18, 6, 13, 7

↑ ↑

start min

↑

min

-2, 4, 5, 6 ⁞ 18, 12, 13, 7

↑ ↑ ↑

start min min

min

-2, 4, 5, 6, 7 ⁞ 12, 13, 7

start min

-2, 4, 5, 6, 7, 12 ⁝ $\underset{\text{start}}{\underset{\uparrow}{13}}$, 18

-2, 4, 5, 6, 7, 12, 13, 18 ↑ min

Time complexity:-

Best case :- $O(n^2)$

Avg case :- $O(n^2)$

Worst case :- $O(n^2)$

) Apply merge sort & order the list of 8 elements Data $d = (45, 67, 12, 5, 22, 30, 50, 20)$ set up a recurrence relation for the number of key comparison made by merge sort.

Given

$d = (45, 67, 12, 5, 22, 30, 50, 20)$.

$\overset{0}{45}, \overset{1}{67}, \overset{2}{-12}, \overset{3}{5}, \overset{4}{22}, \overset{5}{30}, \overset{6}{50}, \overset{7}{20}$

$m = \frac{l+h}{2} = \frac{0+7}{2} = \frac{7}{2} = 3.5 \simeq 4$

$\overset{0}{45}, \overset{1}{67}, \overset{2}{-12}, \overset{3}{5}, \overset{4}{22} \mid \overset{5}{30}, \overset{6}{50}, \overset{7}{20}$

$m = \frac{l+h}{2} = \frac{0+4}{2} = \frac{4}{2} = 2$

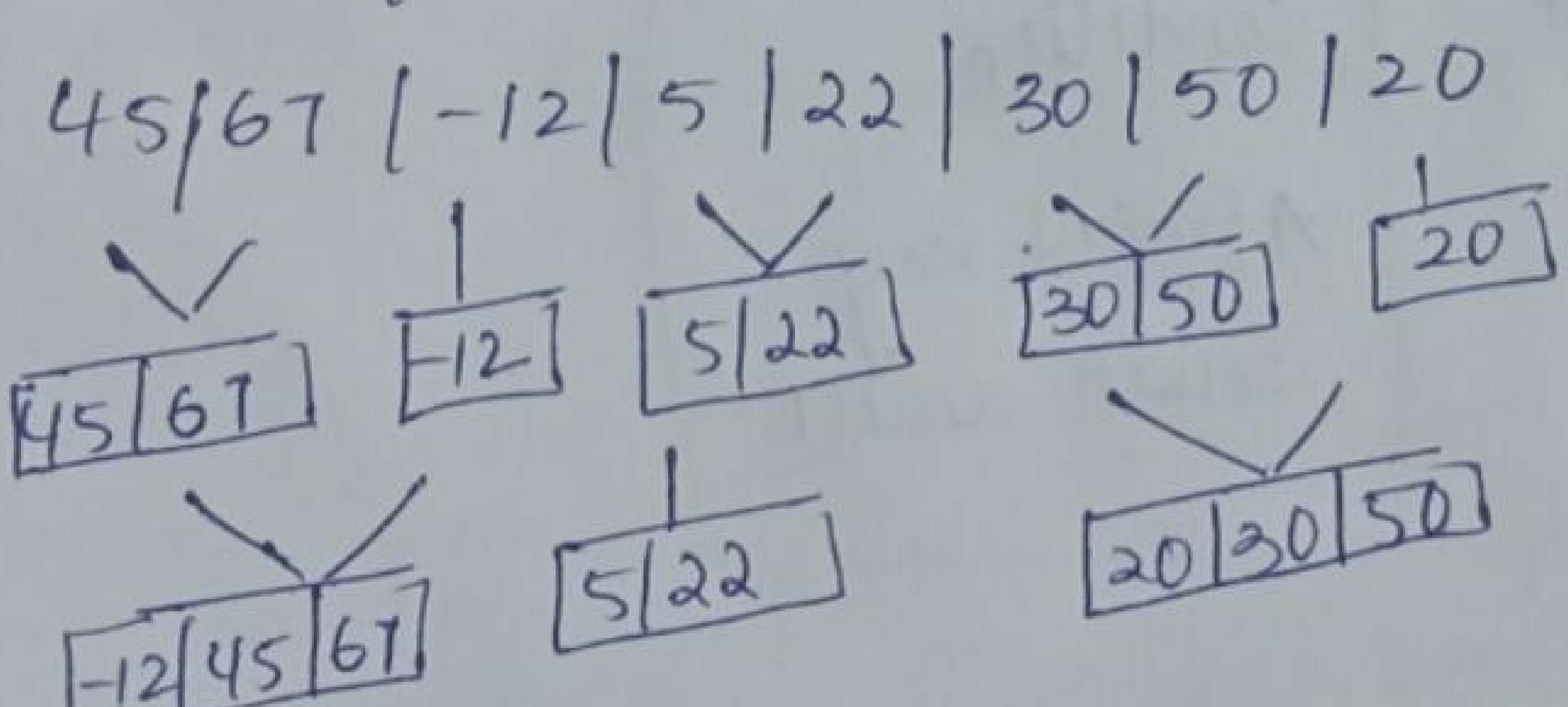
$m = \frac{l+h}{2} = \frac{5+7}{2} = \frac{\cancel{12}^{6}}{\cancel{2}} = 6$

$\overset{0}{45}, \overset{1}{67}, \overset{2}{-12} \mid \overset{3}{5}, \overset{4}{22} \mid \overset{5}{30}, \overset{6}{50} \mid \overset{7}{20}$

$m = \frac{l+h}{2} = \frac{2}{2} = 1$

45, 67 | -12 | 5 | 22 | 30 | 50 | 20

45 | 67 | -12 | 5 | 22 | 30 | 50 | 20



-12	5	22	45	67

20	30	50

-12	5	20	22	30	45	50	67

Recurrence relation:

$T(n) = 2t(n/2) + c(n)$

$a=2,\ b=2,\ k=1,\ p=1$

$\log_2^2 = 1$

$\log_a^b = k$

$p>-1 \Rightarrow \theta(n^k \log_n^{p+1})$

$\theta(n^1 \log_n^{1+1})$

$\theta(n \log_n^2)$

$= \theta(n\log n)$

12) Demonstrate Binary search method to search key=23 from the array arr[]=[2,5,8,12,16,23,38,56,72,91].

Given $arr[] = [\overset{0}{2}, \overset{1}{5}, \overset{2}{8}, \overset{3}{12}, \overset{4}{16}, \overset{5}{23}, \overset{6}{38}, \overset{7}{56}, \overset{8}{72}, \overset{9}{91}]$

key = 23

$B = \frac{0+9}{2} = \frac{9}{2} = 4.5 \simeq 5$

$\overset{0}{2}, \overset{1}{5}, \overset{2}{8}, \overset{3}{12}, \overset{4}{16}, \underset{\substack{\uparrow \\ \text{min}}}{\boxed{\overset{5}{23}}}, \overset{6}{38}, \overset{7}{56}, \overset{8}{72}, \overset{9}{91}$

l	h	mid	condition
0	9	5	A[mid]=key return result

an array of {4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, -9} integers, find max and min product that can be obtained by multiplying 2 integers from array.

Given

arr [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9]

Maximum no's:-

Two max no's = 11, 10 = 11 × 10 = 110

(largest)

Two smallest no's = -9 × -8 = 72

Product = 110 is the highest (maximum).

Minimum no's:-

Two min no's = 11 × -9 = -99

Two min no's = 11 × -8 = -88

product = -99 is the (minimum).

10) solve the following recurrence relations and find the order of growth for solution. $T(n) = 4T(n/2) + n^2$, $T(1) = 1$

$a = 4$.

$b = 2$, $k = 2$, $p = 1$

$$\log_b^a \Rightarrow \log_2^4 \Rightarrow \log_2^{2^2} = 2(1) = 2$$

$$\log_b^a = k$$

$$p \geq -1 \Rightarrow \theta(n^k \log_n^{p+1})$$

$$\theta(n^2 \log n^2) \Rightarrow \theta(n^2 \log n).$$

9) Determine whether $h(n) = n\log n + n$ is in $\Theta(n\log n)$ prove rigorous proof for your conclusion.

Given,

$h(n) = n\log n + n \in \Theta(n\log n)$

$f(n) = n\log n + n$ and $g(n) - n\log n$.

Upper bound :-

$n\log n + n \leq c_2 n\log n$.

$n\log n + n \leq n\log n + n\log n = 2n\log n$

$c_2 = 2$

Lower bound :-

$n\log n + n \geq c_1 n\log n$

$n\log n + n \geq n\log n$

$c_1 = 1$.

Conclusion :-

$n\log n \leq n\log n + n \leq 2n\log n$

for all $n \geq n_0$

$h(n) = n\log n + n$ is in $\Theta(n\log n)$

8) Let $f(n) = n^3 - 2n^2 + n$ and $g(n) = n^2$ show that whether $f(n) = (\Omega\, g(n))$ is true or false & justify your answer ?,

Given

$f(n) = n^3 - 2n^2 + n$; $g(n) - n^2$

$f(n) \geq c \cdot g(n)$

$n^3 - 2n^2 + n \geq c(n^2)$

$n^3 - (2+c)n^2 + n \geq 0$

Choosing constants :-

$n^3 \geq (2+c)n^2$

$n \geq 2+c$

Choosing $c=1$

$n^3 - 3n^2 + n \geq 0$

Conclusion :-

$f(n) = n^3 - 2n^2 + n \in \Omega(g(n))$ //

Big thetha Notation : Determine whether $h(n) = 4n^2+3n$ is $\Theta(n^2)$ or not

Given

$h(n) = 4n^2 + 3n$ is $\Theta(n^2)$

$h(n) \geq c.n^2$

$h(n) = 4n^2 + 3n = n^2(4 + \frac{3}{n})$

we need $n^2(4+\frac{3}{n}) \geq c.n^2$

$n^2(4+\frac{3}{n}) \geq c.n^2$

$4 + \frac{3}{n} \geq c$

$h(n) \geq c.n^2$ for all n

$h(n) \neq \Theta(n^2)$

Big Omega Notation : prove that $g(n) = n^3+2n^2+4n$ is $\Omega(n^3$

Given,

$g(n) = n^3 + 2n^2 + 4n$ is $\omega(n^3)$

$c.n^3 \geq h(n)$

$g(n) = n^3 + 2n^2 + 4n$

$= n^2(n+2) + 4n$

$g(n) = n^2(n+2) + 4n \geq c.n^3$

$n^2(n+2)+4n \geq c.n^3$

$n^2(n+2)+4n \geq c.n^3$

$n^2(n+2)+4n-c.n^3 \geq 0$

$n^2(n+2)+4n-c.n^3 \geq 0$

we need:-

$n^2(n+2)+4n-cn^3 \geq 0$

$0.n^2(n+2)+4n-c.n^3 \geq 0$

$g(n) \geq c.n^3$ for all n

$g(n) \neq \Omega(n^3)$

$\therefore g(n)=n^3+2n^2+4n$ is not $\Omega(n^3)$.

⑤ Big O notation: $f(n)=n^2+3n+5$ is $O(n^2)$.

$f(n) \leq c.n^2$

$f(n)=n^2+3n+5$

$=n^2+3n+5$

$f(n)=n^2+3n+5 \leq c.n^2$

$n^2+3n+5 \leq c.n^2$

$n^2+3n+5 \leq c.n^2$

$3n+5 \leq c.n^2$

$f(n)=n^2+3n+5 \in O(n^2)$

$f(n)=n^2+3n+5 \in O(n^2)$

$$T(n)=\begin{cases} 2T(n-1) & \text{if } n>0 \\ 1 & \text{other wise.} \end{cases}$$

$T(n)=2T(n-1)$

$T(n-1)=2[2T(n-2)]$

$=2^2T(n-2)$

$T(n) = 2^2 [2T(n-3)]$

$= 2^3 T(n-3)$

$T(n) = 2^k + (n-k)$

$n-k=0, \; n=k$

$T(0)=1$

$$\boxed{T(n) = O(2^n)}$$

3) $T(n) = \begin{cases} 2T(n/2)+1 & \text{if } n>1 \\ 1 & \text{otherwise.} \end{cases}$

$T(n) = aT(n/b) + f(n)$

if $f(n) = O(n \log_b^a - c)$

then $T(n) = \theta(n \log_b^a)$

if $f(n) = \theta(n \log_b^a \log_n^{k+1})$

then $T(n) = \theta(n \log_b^a \} \log_n^{k+1})$

if $f(n) = \Omega(n \log_b^a + \epsilon)$

then $T(n) = \theta(f(n))$

$T(n) = 2T(n/2) + 1$

$a=2, \; b=2$

$k=1, \; P=1$

$\log_a^b = \log_2^2 = 1$

$\log_a^b = k$

$P \geq -1 \Rightarrow \theta(n^k \log_n^{p+1})$

$\theta(n^1 \log_n^2)$

$\theta(n \log n)$

① If $t_1(n) \in O(g'(n))$ and $t_2(n) \in O(g^2(n))$, then $t_1(n) + t_2(n) \in$ $O(\max(g'(n)), g_2(n))$, prove that assertions.

Given,

$t_1(n) \leq c_1 \cdot g_1(n)$

$t_2(n) \leq c_2 g_2(n)$

consider $t_1(n) + t_2(n)$

$t_1(n) + t_2(n) \leq c_1 \cdot g_1(n) + c_2 \cdot g_n(n)$

Find an upper bound for $t_1(n) + t_2(n)$,

$\max\{g_1(n), g_2(n)\} \geq g_1(n)$ and

$\max\{g_2(n), g_2(n)\} \geq g_2(n)$

Therefore,

$t_1(n) + t_2(n) \leq c_1$

$\max\{g_1(n), g_2(n)\} + c_2$

$\max\{g_1(n), g_2(n)\}$

Let $c = c_1 + c_2$ then

$t_1(n) + t_2(n) \leq c$

$\max\{g_1(n), g_2(n)\}$

Thus, $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$

$|t_1 tn) + t_2(n)| \leq c$

$\max\{g_1(n), g_2(n)\}$

Thus, the statement is proven.