CSA - 0666

Analytical Problems - II

K. Prajith
192324013
B. Tech AI DS

1) If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$. P.T $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$.

Sol: For any four arbitary real numbers, a_1, a_2, b_1, b_2

Such that, $a_1 \le b_1$ and $a_2 \le b_2$

we have, $a_1 + a_2 \le \text{Max}\{b_1, b_2\}$

Since $t_1(n) \in O(g_1(n))$, then there exists some constant c_1 and non-negative integer n_1 such that.

$$t_1(n) \le c_1 g_1(n) \quad \text{for all } n \ge n_1$$

Since $t_2(n) \in O(g_2(n))$ then there exists some constant c_2 and non-negative integer n_2 such that.

$$t_2(n) \le c_2 g_2(n) \quad \text{for all } n \ge n_2$$

Let $c_3 = \max\{c_1, c_2\}$ and $n = \max\{n_1, n_2\}$

$$\begin{aligned} t_1(n) + t_2(n) &\le c_1 g_1(n) + c_2 g_2(n) \\ &\le c_3 g_1(n) + c_3 g_2(n) \\ &\le c_3 \{g_1(n) + g_2(n)\} \\ &\le 2 c_3 \max\{g_1(n), g_2(n)\} \end{aligned}$$

Hence, $t_1(n) + t_2(n) \in O[\max\{g_1(n), g_2(n)\}]$ with constants c and n_0 required by the O definition being $2c_3 = 2\max\{c_1, c_2\}$ and $\max\{n_1, n_2\}$ respectively.

2. Find the time complexity for the below recurrence relation.

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

By using Master's Theorem,

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

$f(n) = n^k \log_n^p$

Here,

$a=2 \quad b=2 \quad k=0$

$\log_b^a = \log_2^2 = 1$

$\log_b^a > k$

Case(i) $O(n) = O(n \cdot \log_b^a)$

$= O(n \cdot 1)$

$= O(n)$

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

By using Backward substitution,

$T(n) = 2T(n-1)$ —①

Sub $n = n-1$

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

$T(n-1) = 2T(n-2)$ —②

Sub ② in ①

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

$T(n) = 2^2 T(n-2)$ —③

Sub $n = n-2$

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

$T(n-2) = 2T(n-3)$ —④

Sub (4) in (3)

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

$T(n) = 2^3 T(n-3)$ — (5)

Sub $n = n-3$

$T(n-3) = 2T(n-3-1)$

$T(n-3) = 2T(n-4)$

Sub (6) in (5)

$T(n) = 2^3 [2T(n-4)]$

$= 2^4 T(n-4)$ — (7)

Sub $n = n-4$

$T(n-4) = 2T(n-4-1)$

$T(n-4) = 2T(n-5)$ — (8)

Sub (8) in (7)

$T(n) = 2^4 [2T(n-5)]$

$T(n) = 2^5 T(n-5)$

By generalising, we get

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

$T(0) = 0$

$n-k = 0$

$n = k$

If $T(0) = 1$

$T(n) = 2^k \cdot T(0)$

$T(n) = 2^k$ $\quad \because n = k$

$\therefore T(n) = O(2^n)$

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

Big O condition: $f(n) \leq g(n).c$

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

for $n \geq 1$, $n^2 \geq n$

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

$$f(n) = n^2 + 3n + 5 < 9n^2 \text{ for } n \geq 1$$

for $c = 9$ and $n_0 = 1$

$$f(n) \leq c.n^2 \text{ for all } n \geq n_0$$

∴ which stated that $f(n)$ is $O(n^2)$.

6) Big Ω notation: P.T $g(n) = n^3 + 2n^2 + 4n$ is $\Omega(n^3)$

Big Ω condition : $f(n) \geq g(n)$

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

for $n \geq 1$

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

Since $2n^2$ & $4n$ are less than n^3 when $n \geq 1$

So, $c = 1$ & $n_0 = 1$

$$g(n) = c.n^3 \text{ for all } n \geq n_0$$

∴ which stated that $g(n)$ is $\Omega g(n^3)$

Big O notation: $h(n) = 4n^2 + 3n$ is $O(n^2)$ or not.

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

for $n \geq 1$, $h(n) \leq 4n^2 + 3n^2$

since $3n$ is less than n^2 when $n \geq 1$

$h(n) \leq 7n^2$ for $n \geq 1$

$\therefore h(n)$ is $O(n^2)$

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

for $(n) \geq 1$ $h(n) \geq 4n^2$

since $3n$ is positive.

$h(n)$ is $\Omega(n^2)$

$\therefore$ Since $h(n)$ is $O(n^2)$ & $\Omega(n^2)$ it is $\Theta(n^2)$

8. Let $f(n) = n^3 - 2n^2 + n$ & $g(n) = -n^2$ show wether $f(n) = \Omega(c \cdot g(n))$ is true or false. Justify your answer.

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

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

$n = 1$

$1^3 - 2(1)^2 + 1 \geq -(1)^2$

$1 - 2 + 1 \geq 1$

$0 \geq 1$ ✗

$n = 2$

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

$8 - 8 + 2 \geq 4$

$2 \geq 4$ ✗

$n = 3$

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

$27 - 18 + 3 \geq 9$

$12 \geq 9$ ✓

$n=4$

$$4^3 - 2(4)^2 + 4 \geq -(4)^2$$

$$64 - 32 + 4 \geq 16$$

$$36 \geq 16$$ ✓

$n=5$

$$5^3 - 2(5)^2 + 4 \geq (-5)^2$$

$$125 - 50 + 4 \geq 25$$

$$79 \geq 25$$ ✓

So, $f(n) \geq g(n)$ for $n > 2$

So, it is the best case according to the asymptotic notation.

9. Determine wether $h(n) = n\log n + n$ is $\Theta(n\log n)$ prove it.

① Upper Bound:

We need to find $c_1 \& n_0$

$h(n) \leq c_1 . n\log n$ for all $n \geq n_0$

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

$h(n) \leq n\log n + n\log n$

$h(n) \leq 2n\log n$

Now, let, $c_1 = 2$ then $h(n) \leq 2n\log n$ for all $n \geq 1$

So, $h(n)$ is $O(n\log n)$

② Lower Bound:

We need to find $c_1 \& n_0$

$h(n) \geq c_2 . n\log n$ for all $n \geq n_0$

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

$h(n) \geq \frac{1}{2} . \log n$ (for $n \geq 2$)

let $c_2 = \frac{1}{2}$ then $h(n) \geq \frac{1}{2} \cdot n\log n$. for all $n \geq 2$

So $h(n)$ is $\Omega(n\log n)$

Since $h(n)$ is $O(n\log n)$ & $\Omega(n\log n)$.

we consider and justify it as $\Theta(n\log n)$.

10. Solve the recurrence relation & find the order of growth.

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

By using Master Theorem,

$$T(n) = aT(n/b) + f(n) \qquad f(n) = n^k \log_b^p$$

Here,

$a = 4 \quad b = 2 \quad k = 2 \quad p = 0$

$$\log_b^a = \log_2^4 = 2$$

$$\log_b^a = k$$

$p > -1$

$$\Rightarrow O(n^k \log_n^{p+1})$$

$$= O(n^2 \log_n^{0+1})$$

$$= O(n^2 \log n)$$

∴ The order of growth for the solution is $n^2 \log n$.

11. Given an array [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9]. Find the max & min product that can be obtained by multiplying two integers from the array.

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

Max no = 11, 10

Product = 110

Min no = -9, -8 $\quad\quad$ -9×11

$= -72$ $\quad\quad$ $= -99$

∴ Min-Product = -99

Max-Product = 110

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

0	1	2	3	4	5	6	7	8	9
2	5	8	12	16	23	38	56	72	91

low = 0

high = 9

$mid = \frac{low + high}{2}$

$= \frac{0+9}{2} = 4$

a[mid] == key

a[4] == 23

16 != 23

16 < 23

low = mid + 1

5	6	7	8	9
23	38	56	72	91

low = 5 high = 9

$mid = \frac{5+9}{2} = \frac{14}{2} = 7$

a[mid] == key

a[7] == 23

56 != 23 56 > 23

high = mid - 1

5	6	7
23	38	56

low = 5 high = 7

$mid = \frac{5+7}{2} = 6$

a[6] == 23

38 != 23

38 > 23

high = mid - 1

5
23

low = 5 high = 5

$mid = \frac{5+5}{2} = 5$

a[5] == key

23 == 23 ✓

Return the position of the key (i.e) 5

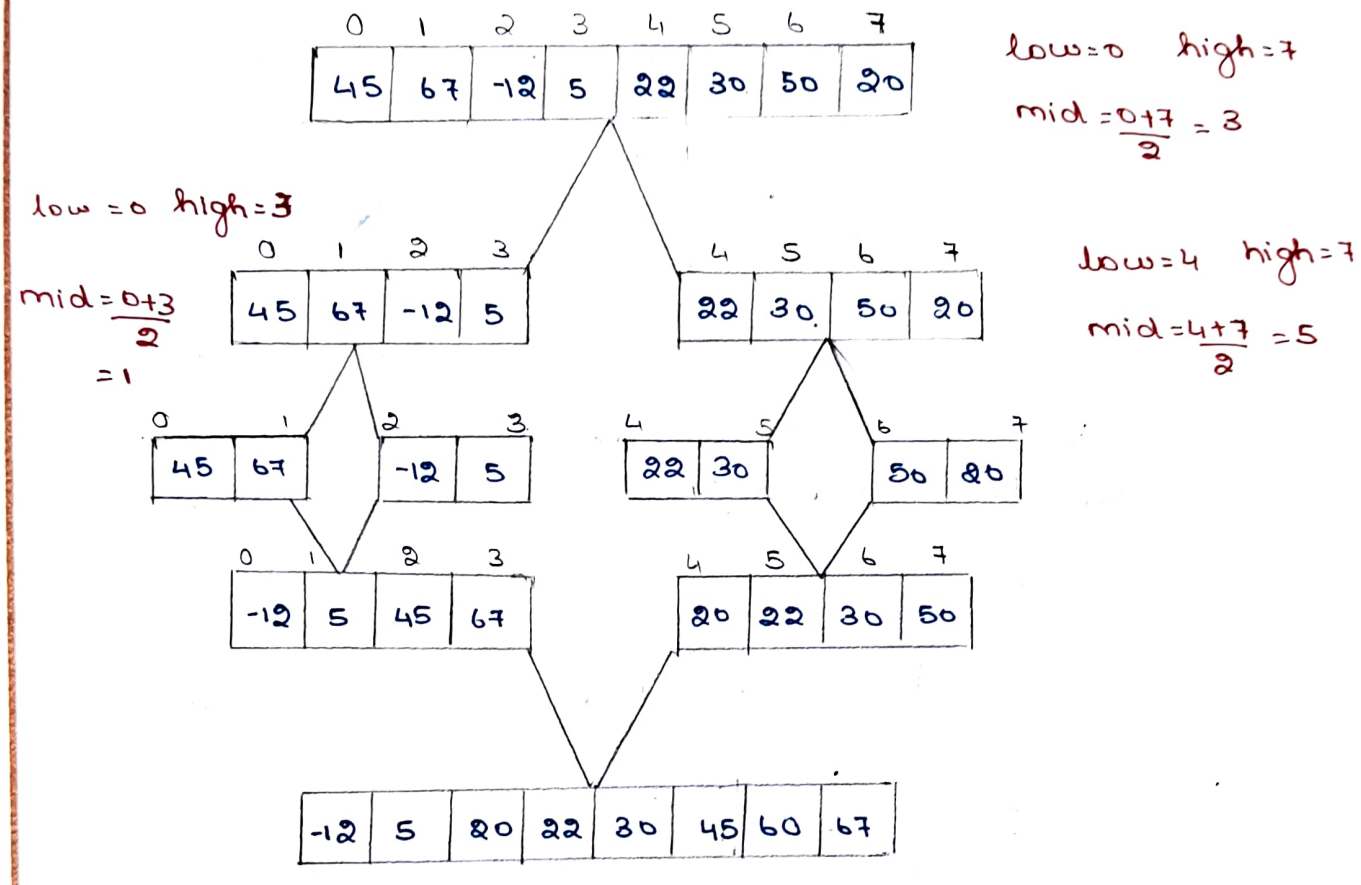
Pseudocode:

```
binary-search (a, n, key):
    low = 0
    high = n-1
    while (low <= high):
        mid = (high + low) // 2
        if a[mid] == key:
            return mid
        a[mid] > key
            high = mid-1
        a[mid] < key
            low = mid+1
    return -1 [If not found].
```

Time Complexity:

O(nlogn).

Apply merge sort and order the list of 8 elements d = [45, 67, -12, -12, 5, 22, 30, 50, 20]. Set up a recurrence relation for the number of key comparisons made by merge sort.



∴ The sorted list is :

-12, 5, 20, 22, 30, 45, 60, 67.

Time Complexity: O(nlogn)

Recurrence relation: ~~O(nlogn)~~ $T(n) = 2T(n/2) + (n-1)$

Recurrence relation: $T(n) = 2T(n/2) + (n-1)$ —①

Sub $n = n/2$

$$T(n/2) = 2T\left(\frac{n}{2^2}\right) + \left(\frac{n}{2} - 1\right) \quad —②$$

Sub ② in ①

$$T(n) = 2\left[2T\left[\frac{n}{2^2}\right] + \left[\frac{n}{2} - 1\right]\right] + (n-1)$$

$$= 2^2 T\left[\frac{n}{2^2}\right] + n - 2 + n - 1$$

$$T(n) = 2^2 T\left[\frac{n}{2^2}\right] + 2n - 3 \quad —③$$

Sub $n = n/4$

$$T(n/4) = 2T\left(\frac{n}{2^3}\right) + \left[\frac{n}{2^2} - 1\right] \quad —④$$

Sub ④ in ③

$$T(n) = 2^2 T\left[2T\left[\frac{n}{2^3}\right] + \frac{n}{2^2} - 1\right] + 2n - 3$$

$$= 2^3 T\left[\frac{n}{2^3}\right] + n - 1 + 2n - 3$$

$$= 2^3 T\left[\frac{n}{2^3}\right] + 3n - 4$$

By generalizing,

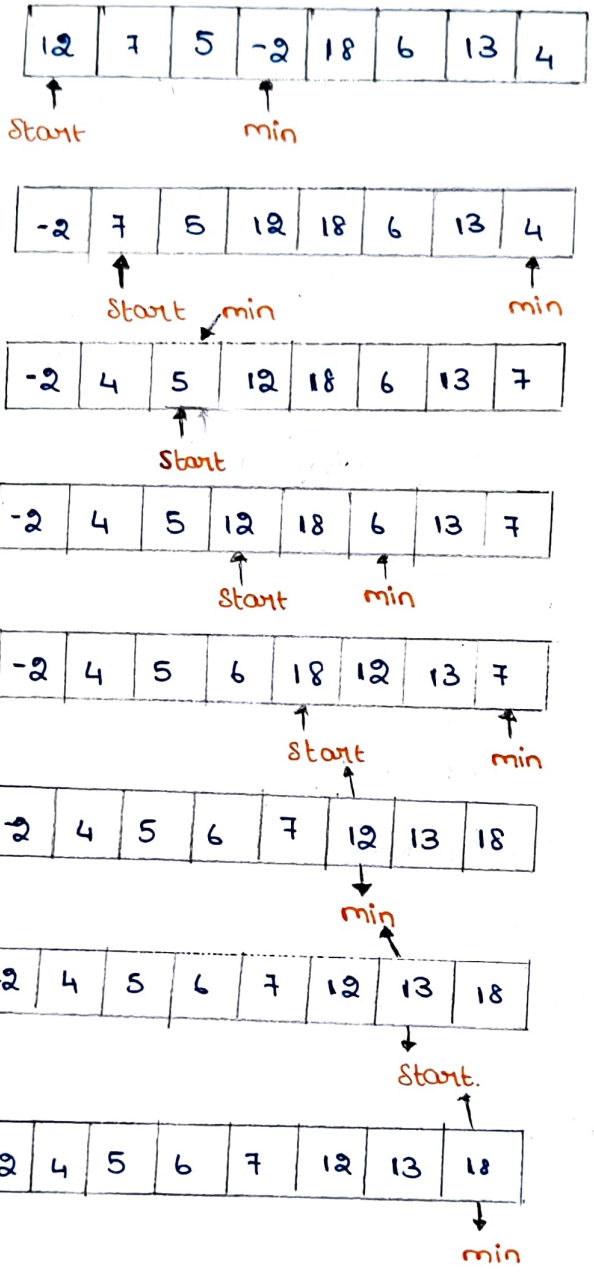
$$T(n) = 2^k T\left[\frac{n}{2^k}\right] + kn - k + 1$$

$\frac{n}{2^k} = 1 \quad k = \log_2 n$

$$T(n) = n \log_2 n - n + 1$$

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

Find the no. of times to perform swapping for selection sort. Also estimate the time complexity. S= (12, 7, 5, -2, 18, 6, 13, 4)

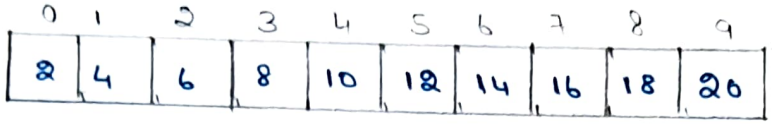


Sorted list: -2, 4, 5, 6, 7, 12, 13, 18

Usually, the number of swaps required will be n-1. But for this question there are only 4 swaps.

Time Complexity: $O(n^2)$. It is n^2 in all the three cases.

15. Find the index of the target value 10 using binary search from the following list of elements. [2, 4, 6, 8, 10, 12, 14, 16, 18, 20].



low = 0 high = 9

mid = $\frac{0+9}{2}$ = 4

a[mid] == key

a[4] == 10

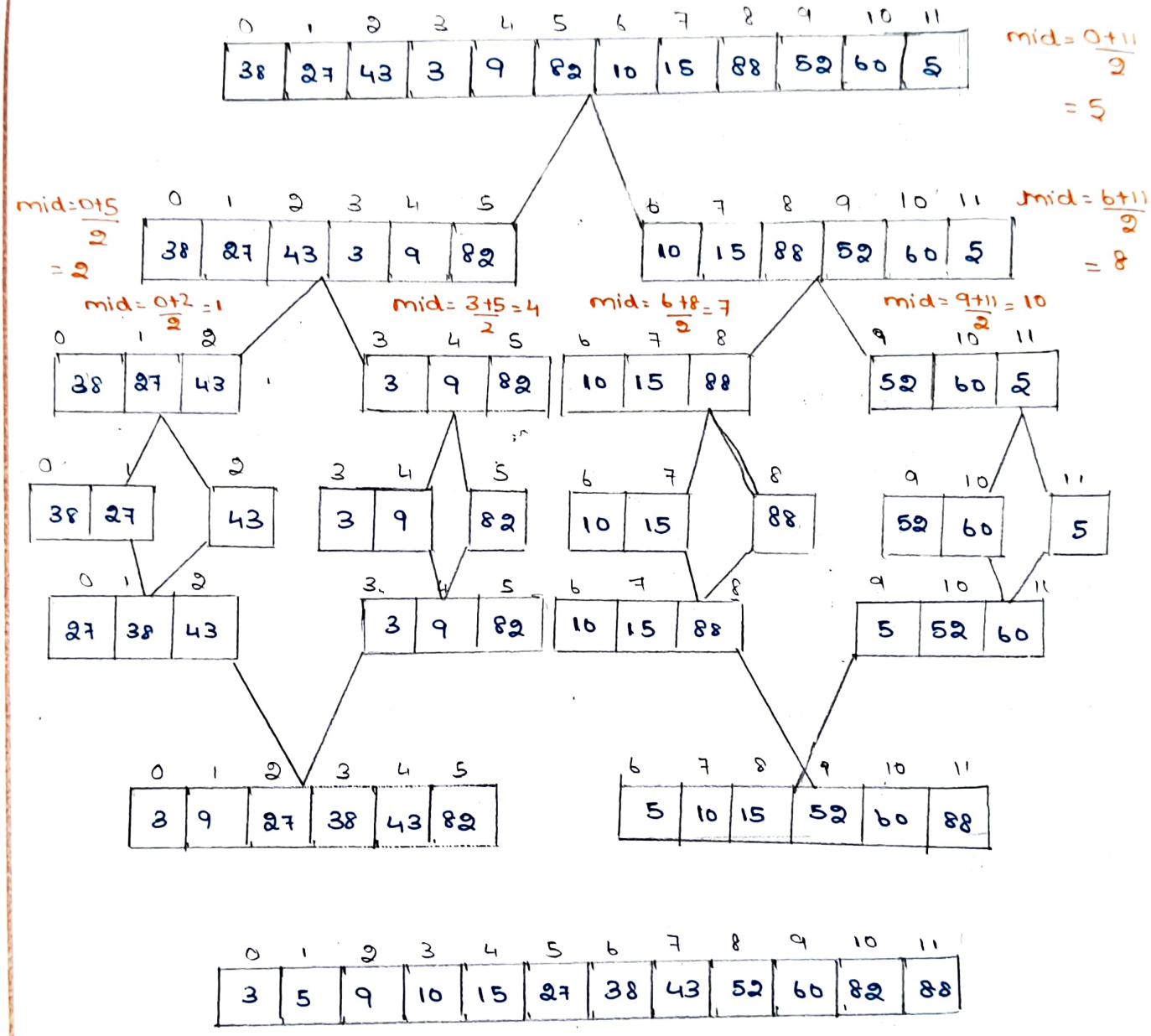
10 == 10 ✓

Return the position of the key (i.e) 4

Pseudocode:

```
binary-search (array, size of array, key)
    low = 0
    high = size - 1
    while (low <= high)
        mid = (high + low) / 2
        if a[mid] == key
            return mid
        a[mid] > key
            high = mid - 1
        a[mid] < key
            low = mid + 1
    return -1  [If not found]
```

Solve the elements using Merge sort divide & conquer strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] & analyze the time complexity.



∴ The sorted list is: 3, 5, 9, 10, 15, 27, 38, 43, 52, 60, 82, 88.

Pseudocode:

```
partition (low, high)
    if l<h:
      mid = h+l/2
      partition (l, mid)
      partition (mid+1, h)
      merge (l, mid, h)
    end if
```

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

By using Master theorem.

$a=2 \quad b=2 \quad k=1$

$\log_b^a = \log_2^2 = 1$

Comparing $\log_a^b$ & k

$\log_a^b = k$

∴ case (ii)

$P<-1$

∴ $O(n^k \log^{p+1} n)$

$= O(n \log n)$

∴ Time complexity: $O(n \log n)$

Sort the array 64, 34, 25, 12, 22, 11, 90 using Bubble sort. What is the time complexity of Selection sort in Best, average & worst case)

64 34 25 12 22 11 90

34 64 25 12 22 11 90

34 25 64 12 22 11 90 — Phase - I

34 25 12 64 22 11 90

34 25 12 22 64 11 90

34 25 12 22 11 64 90

34 25 12 22 11 64 90

---

34 25 12 22 11 64 90

25 34 12 22 11 64 90

25 12 34 22 11 64 90 — Phase - II

25 12 22 34 11 64 90

25 12 22 11 34 64 90

25 12 22 11 34 64 90

---

25 12 22 11 34 64 90

12 25 22 11 34 64 90

12 22 25 11 34 64 90 — Phase - III

12 22 11 25 34 64 90

12 22 11 25 34 64 90

---

12 22 11 25 34 64 90

12 22 11 25 34 64 90 — Phase - IV

12 11 22 25 34 64 90

12 11 22 25 34 64 90

12 11 22 25 34 64 90

11 22 22 25 34 64 90 - Phase V

11 12 22 25 34 64 90

---

11 12 22 25 34 64 90

11 12 22 25 34 64 90 - Phase VI

∴ The sorted list: 11, 12, 22, 25, 34, 64, 90

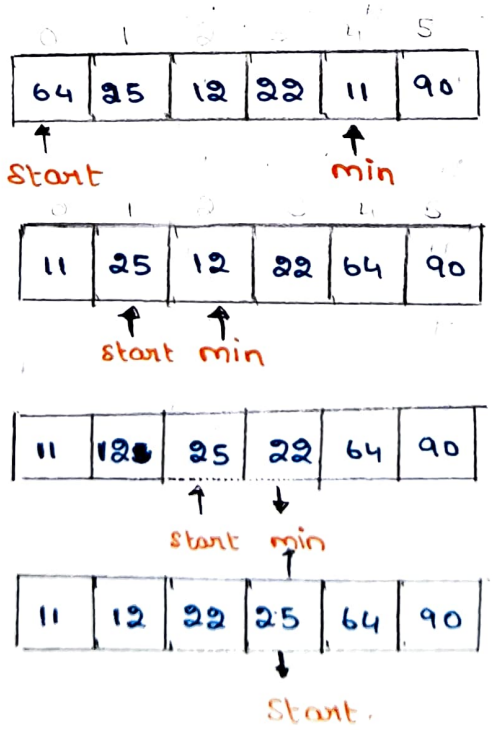
Time Complexity: $O(n^2)$

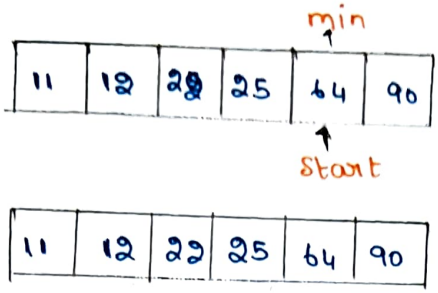
Selection sort: Best Case: $O(n^2)$

Worst Case: $O(n^2)$

Average case: $O(n^2)$

18. Sort the array 64, 25, 12, 22, 11 using selection sort. What is the time complexity.





The sorted list is : 11, 12, 22, 25, 64, 90

Time Complexity:

The time complexity of the Selection Sort Algorithm is always $O(n^2)$. Because, for each element in the array, the algorithm iterate throughs the remaining elements to find the minimum.

The outer loop runs $n+1$ times and the inner loops runs $n-1, n-2$ till 1 time.

Best case: $O(n^2)$

Wort case: $O(n^2)$

Average case: $O(n^2)$

19. Solve the following using Insertion sort using Brute-force approach & analyze the complexity

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

3 9 10 15 27 38 43 52 60 82 88

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

∴ The sorted list is: 3, 5, 9, 10, 15, 27, 38, 43, 52, 60, 82, 88.

Pseudocode:

```
Insertion_sort (a, size of a):
    for i in range (2, size):  ---> (n-2)
        key = a[i]  - (1)
        j = i - 1  - (1)
        while j>0 and a[j]> key  n*(n-2)
            a[ j+1] = a[j]  - (1)
            j-=1  - (1)
        a[i+1] = key  - (1)
    return a  - (1)
```

Time complexity:

$$T(n) = n^*(n-2) + n-2+1+1+1+1+1+1+1$$

$$= n^2 - 2n + n - 2 + 6$$

$$= n^2 - n + 4$$

$\therefore O(n) = O(n^2)$

Time complexity is $O(n^2)$

Given 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 using insertion sort and analyze the time complexity of the algorithm.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

∴ The sorted list is:

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

Pseudo code:

```
Insertion-sort (a, size of a):
    for i in range (2, size-1):
        key = a[i]
        j = i-1
        while j > 0 & a[j] > key
            a[j+1] = a[j]
            j -= 1
        a[i+1] = key
    return array.
```

Time complexity: $O(n^2)$