1. Consider the following Python code:

def total\_function(arr):

total = 0

for i in range(len(arr)):

for j in range(i, len(arr)):

total += arr[i] + arr[j]

return total

What is the time complexity of the ‘total\_function’ in terms of Big O notation?

1. O(n)
2. O(n^2)
3. O(1)
4. O(log n)

Correct answer: b

Explanation: The code consists of the nested loops. The outer loops runs for len(arr) times and inner loop runs from i to len(arr) for each iteration of the outer loop. This results in a time complexity of O(n^2).

2. What does it mean when we say an algorithm has an “exponential” time complexity O(2^n)?

1. It runs in constant time
2. It is highly efficient and fast
3. It is the same as linear time complexity O(n)
4. It becomes significantly slower as the input size increases

Correct answer: d

3. Consider the following Python code:

def selection\_sort(arr):

for i in range(len(arr)):

min\_index = i

for j in range(i+1, len(arr)):

if arr[j] < arr[min\_index]:

min\_index = j

arr[i], arr[min\_index] = arr[min\_index], arr[i]

What is the time complexity of the ‘selection\_sort’ function in terms of Big O notation?

1. O(n)
2. O(log n)
3. O(1)
4. O(n^2)

Correct answer: d

The code implements selection sort algorithm and consists of two nested loops. Thus, the overall time complexity is O(n^2)

4. Consider the following Python code:

def iterative\_function():

i = n

while(i>2):

i = i^(1/25)

What is the time complexity of the ‘iterative\_function’ in terms of Big O notation?

1. O(log\_2(log\_25 n))
2. O(log\_25(log\_2n)
3. O(log\_2(log\_2n))
4. O(logn)

Correct answer: b

Explanation: Refer Example 6 in the Lecture 1 Live Session

5. What is the time complexity of a basic loop that iterates through an array of size ‘n’ and performs a single operation for each element?

1. O(1)
2. O(n)
3. O(log n)
4. O(n^2)

Correct answer: b

Explanation: To iterate through an array of size ‘n’, it takes O(n) as time complexity.

6. Which of the following time complexities represents an algorithm that runs in constant time regardless of the input size?

1. O(n)
2. O(log n)
3. O(1)
4. O(n^2)

Correct answer: c

Explanation: O(1) represents the constant time complexity.