University of Colombo School of Computing
SCS 1304 - Problem Solving Strategies and Computation Approaches

Lab Sheet 07

Analysis of Algorithms

01. Consider the following code snippet and write the time complexity (Big O) of each code.

A.

```
FOR i FROM 0 TO n-1
  FOR j FROM 0 TO i
    PRINT i, j
```

B.

```
FOR i FROM 0 TO n-1
  FOR j FROM 0 TO m-1
    FOR k FROM 0 TO p-1
      PRINT i, j, k
```

C.

```
FOR i FROM 0 TO n-1
  FOR j FROM 0 TO i
    IF (i % 2 == 0)
      PRINT i, j
    ELSE
      FOR k FROM 0 TO j
        PRINT i, j, k
```

D.

```
i = 1
WHILE i < n
  FOR j FROM 0 TO i
    PRINT i, j
  i = i * 2
```

E.

```
FOR i FROM 1 TO n
  FOR j FROM 1 TO i * i
    IF (j % i == 0)
      FOR k FROM 0 TO j
        PRINT i, j, k
```

F.

```
FOR i FROM 0 TO n-1
  k = i
  WHILE k < n
    PRINT i, k
    k = k + i + 1
```

2. Given an integer n, write a function to find the n-th Fibonacci number using dynamic programming and explain the time complexity of the function.

```
FUNCTION FibonacciDP(n):
        IF n <= 1:
                RETURN n
        fib = [0] * (n + 1)
        fib[1] = 1
        FOR i FROM 2 TO n:
                fib[i] = fib[i - 1] + fib[i - 2]
        RETURN fib[n]
```

a. Test your function with the integer 10 to find the 10th Fibonacci number.
b. Explain the time complexity of the FibonacciDP function.
c. What is the space complexity of this algorithm? Compare it with non- DP fibonacci function.

Quizzes

1. The Worst case occurs in the linear search algorithm when _____
   a) Item is somewhere in the middle of the array
   b) Item is not in the array at all
   c) Item is the last element in the array
   d) Item is the last element in the array or is not there at all

2. Which is used to measure the Time complexity of an algorithm Big O notation?
   a) describes the limiting behavior of the function
   b) characterizes a function based on the growth of the function
   c) upper bound on the growth rate of the function
   d) all of the mentioned

3. Which is used to measure the Time complexity of an algorithm Big O notation?

a) describes the limiting behavior of the function
b) characterizes a function based on the growth of the function
c) upper bound on the growth rate of the function
d) all of the mentioned

4. If for an algorithm, the time complexity is given by O(n) then the complexity of it is _____

a) constant
b) linear
c) exponential
d) none of the mentioned

5. If for an algorithm, the time complexity is given by O(n2) then the complexity will _____

a) constant
b) quadratic
c) exponential
d) none of the mentioned

6. If for an algorithm, the time complexity is given by O((3⁄2)n) then the complexity will be _____

a) constant
b) quadratic
c) exponential
d) none of the mentioned

7. What is the time and space complexity of the following code?

```
a = 0
b = 0
for i in range(N):
  a = a + random()

for i in range(M):
  b= b + random()
```

1) O(N * M) time, O(1) space
2) O(N + M) time, O(N + M) space
3) O(N + M) time, O(1) space
4) O(N * M) time, O(N + M) space

8. What is the time complexity of the following code:

```
k = 0;
for i in range(n//2,n):
    for j in range(2,n,pow(2,j)):
    k = k + n / 2;
```

1) O(n)
2) O(N log N)
3) O(n^2)
4) O(n^2Logn)

9. Algorithms A and B have a worst-case running time of O(n) and O(logn), respectively. Therefore, algorithm B always runs faster than algorithm A.

1) true
2) false

10. Which of the following best describes the useful criterion for comparing the efficiency of algorithms?

1) Time
2) Memory
3) Both of the above
4) None of the above

11. How is time complexity measured?

1) By counting the number of algorithms in an algorithm.
2) By counting the number of primitive operations performed by the algorithm on a given input size.
3) By counting the size of data input to the algorithm.
4) None of the above

12. What will be the time complexity of the following code?

```
for i in range(n):
    modified_i = i * k
```

1) O(n)
2) O(k)
3) O(logkn)
4) O(lognk)

13. If for an algorithm the time complexity is given by O(n log n), then the complexity will be

a) linear
b) logarithmic
c) quadratic
d) exponential

14. If for an algorithm the time complexity is given by O(n log n), then the complexity will be

a) linear
b) quadratic
c) linearithmic
d) cubic

15. What is the time and space complexity of the following code?

```
a = [0] * N
for i in range(N):
    for j in range(N):
        a[i] = a[j] + 1
```

1) O(N^2) time, O(N) space
2) O(N^2) time, O(N^2) space
3) O(N) time, O(N) space
4) O(N^2) time, O(1) space

16. An algorithm has a worst-case running time of O(n^4). If the input size is doubled, the running time of the algorithm will increase by a factor of _____

a) 2
b) 4
c) 8
d) 16

17. Which of the following statements is true?

1) An algorithm with a time complexity of O(n log n) is always more efficient than an algorithm with a time complexity of O(n^2)
2) An algorithm with a time complexity of O(n^2) is always more efficient than an algorithm with a time complexity of O(n log n)
3) An algorithm with a time complexity of O(n log n) can be more efficient than an algorithm with a time complexity of O(n^2), depending on the size of the input
4) None of the above

18. What is the time complexity of the following code?

```
def mystery_function(arr):
    n = len(arr)
    if n == 1:
        return arr[0]
    left_sum = mystery_function(arr[:n//2])
    right_sum = mystery_function(arr[n//2:])
    return left_sum + right_sum
```

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

19. What is the time complexity of the following code?

```
def complex_function(n):
    if n <= 1:
        return 1
    total = 0
    for i in range(n):
        total += complex_function(i)
    return total
```

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

20. What is the time complexity of the following code?

```
def different_function(n):
    i = n
    while i > 0:
        for j in range(i):
            print(j)
        i = i // 2
```

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