1. Which of the following case does not exist in complexity theory?

A. Best case

B. Worst case

C. Average case

D. Null case

ANS: D Null case

Explanation:Null case does not exist in complexity Theory

2. What is the time, space complexity of following code:

int a = 0, b = 0;

for (i = 0; i < N; i++)

{

a = a + rand();

}

for (j = 0; j < M; j++)

{

b = b + rand();

}

A. O(N \* M) time, O(1) space

B. O(N + M) time, O(N + M) space

C. O(N + M) time, O(1) space

D. O(N \* M) time, O(N + M) space

ANS: C o(N+M) time,o(1) space

Explanation:The first loop is O(N) and the second loop is O(M). Since we don’t know which is bigger, we say this is O(N + M). This can also be written as O(max(N, M)).

Since there is no additional space being utilized, the space complexity is constant / O(1).

3. The complexity of linear search algorithm is

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: o(n)

Explanation: The worst case complexity of linear search is O(n)

4.What is the time complexity of following code:

int a = 0;

for (i = 0; i < N; i++)

{

for (j = N; j > i; j--)

{

a = a + i + j;

}

}

A. O(N)

B. O(N\*log(N))

C. O(N \* Sqrt(N))

D. O(N\*N)

ANS: D o(N\*N)

Explanation:= N + (N – 1) + (N – 2) + … 1 + 0

= N \* (N + 1) / 2

= 1/2 \* N^2 + 1/2 \* N

O(N^2) times

5. The Worst case occur in 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

ANS: D. Item is the last element in the array or is not there at all

Explanation:The Worst case occur in linear search algorithm when Item is the last element in the array or is not there at all

6. What is the time complexity of following code:

int i, j, k = 0;

for (i = n / 2; i <= n; i++)

{

for (j = 2; j <= n; j = j \* 2)

{

k = k + n / 2;

}

}

A. O(n)

B. O(nLogn)

C. O(n^2)

D. O(n^2Logn)

ANS: B O(nLogn)

Explanation:Let’s take the examples here.

for n = 16, j = 2, 4, 8, 16

for n = 32, j = 2, 4, 8, 16, 32

So, j would run for O(log n) steps.

i runs for n/2 steps.

So, total steps = O(n/ 2 \* log (n)) = O(n\*logn)

7. The worst case occur in quick sort when

A. Pivot is the median of the array

B. Pivot is the smallest element

C. Pivot is the middle element

D. None of the mentioned

ANS: B Pivot is the smallest element

Explanation:This happens when the pivot is the smallest (or the largest) element. Then one of the partitions is empty, and we repeat recursively the procedure for N-1 elements.

8. What does it mean when we say that an algorithm X is asymptotically more efficient than Y?

A. X will always be a better choice for small inputs

B. X will always be a better choice for large inputs

C. Y will always be a better choice for small inputs

D. X will always be a better choice for all inputs

ANS: B. X will always be a better choice for large inputs

Explanation: An algorithm X is said to be asymptotically better than Y if X takes smaller time than y for all input sizes n larger than a value n0 where n0 > 0.

9. The complexity of Fibonacci series is

A. O(2n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: A O(2n)

Explanation:= Fibonacci is f(n) = f(n-1) + f(n-2), f(0) = 0, f(1) = 1. Let g(n) = 2n. Now prove inductively that f(n) > = g(n).

10. What is the time complexity of following code:

int a = 0, i = N;

while (i > 0)

{

a += i;

i /= 2;

}

A. O(N)

B. O(Sqrt(N))

C. O(N / 2)

D. O(log N)

ANS: D O(log N)

Explanation:We have to find the smallest x such that N / 2^x N x = log(N).

11. What is the time complexity of following code:

int a = 0, i = N;

while (i > 0)

{

a += i;

i /= 2;

}

A. O(N)

B. O(Sqrt(N))

C. O(N / 2)

D. O(log N)

ANS: D. O(log N)

Explanation: We have to find the smallest x such that N / 2^x N x = log(N)

12. The complexity of Binary search algorithm is

A. O(n)

B. O(log )

C. O(n2)

D. O(n log n)

ANS: B.O(n log n)

Explanation: The compexity of binary search is O(logn).

13. The complexity of merge sort algorithm is

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: D. 0(n log n)

Explanation: The worst case complexity for merge sort is O(nlogn).

14. The complexity of Bubble sort algorithm is

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: C. O(n2)

Explanation: The worst case complexity for Bubble sort is O(n2)ans best case is O(n).

15. The worst case complexity for insertion sort is

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: C. o(n2)

Explanation: In worst case nth comparison are required to insert the nth element into correct position.

16. The worst case complexity of quick sort is

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

ANS: C. O(n2)

Explanation: The worst case complexity of quick sort is O(n2).

17. To measure Time complexity of an algorithm Big O notation is used which:

A. describes limiting behaviour of the function

B. characterises a function based on growth of function

C. upper bound on growth rate of the function

D. all of the mentioned

ANS: D. all of the above

Explanation: Big O notation describes limiting behaviour, and also gives upper bound on growth rate of a function.

18. If for an algorithm time complexity is given by O(1) then complexityof it is:

A. constant

B. polynomial

C. exponential

D. none of the mentioned

ANS: A. constant

Explanation: The growth rate of that function will be constant.

19.If for an algorithm time complexity is given by O(log2n) then complexity will:

A. constant

B. polynomial

C. exponential

D. none of the mentioned

ANS: D. none of the above

Explanation: The growth rate of that function will be logarithmic therefore complexity will be logarithmic.

20. If for an algorithm time complexity is given by O(n) then complexityof it is:

A. constant

B. linear

C. exponential

D. none of the mentioned

ANS: B. linear

Explanation: The growth rate of that function will be linear.