/MODULE 2 : REVIEW QUESTIONS 19/20

1. What is recursion in the context of algorithms?

Choices:

- A method that exclusively uses loops to solve problems
- A technique that avoids using functions entirely
- **A process where a function calls itself directly or indirectly ✅**
- A process where a function calls another function repeatedly
  Explanation: A function solves a problem by calling itself

2. Which of the following best describes the difference between iteration and recursion?

Choices:

- Recursion is always faster than iteration
- Iteration uses the stack while recursion does not
- **Iteration uses repetition structures while recursion uses selection structures ✅**
- Recursion consumes less memory than iteration
  Explanation: Loops vs. recursive function with conditions.

3. What is the primary reason for defining a base case in a recursive function?

Choices:

- **To avoid infinite loops ✅**
- To allow multiple functions to be called simultaneously
- To reduce memory usage
- To increase execution speed
  Explanation: Base case stops the recursion.

4. In the context of recursion, what is tail recursion?

Choices:

- When the function calls another function instead of itself
- When the recursive call happens at the beginning of the function
- When there are multiple recursive calls within the same function
- **When the recursive call is the last operation performed in the function ✅**
  Explanation: Nothing is left to do after the call; can be optimized.

5. What is the output of the following recursive function when n=3?

```
int fun(int n){ if(n==1) return 1; else return 1 + fun(n-1); }
```

Choices:

- **3 ✅**
- 4
- 1
- 2
  Explanation: fun(3) = 1 + fun(2) = 3.

6. What is the time complexity of an algorithm that checks each triple in an array of size N?

Choices:

- O(N)
- **$O(N^3)$ ✅**
- O(log N)
- O(N log N)
  Explanation: Three nested loops → cubic growth.

7. Which asymptotic notation represents the worst-case scenario of an algorithm's running time?

Choices:

- Ω (Omega)
- β (Beta)
- Θ (Theta)
- **O (Big Oh)** ✅
  Explanation: Big O shows maximum running time.

8. What is space complexity in the context of algorithm analysis?

Choices:

- The time it takes to compile an algorithm
- The number of CPU cycles needed to execute an algorithm
- **The amount of memory required by an algorithm to run** ✅
- The number of lines of code in an algorithm
  Explanation: Measures memory usage during execution.

9. If an algorithm has a space complexity of O(1), what does this indicate?

Choices:

- The space required depends on the number of recursive calls
- The space required grows exponentially with input size
- **The space required is constant regardless of input size** ✅
- The space required grows linearly with input size
  Explanation: Memory usage does not change with input.

10. What is the time complexity of binary search in a sorted array of size N?

Choices:

- **O(log N)** ✅
- O(N log N)
- O(N²)
- O(N)
  Explanation: Halves search space each step → logarithmic.

11. Which type of recursion occurs when a function calls another function, which in turn calls the original function?

Choices:

- **Indirect recursion** ✅
- Non-tail recursion
- Tail recursion
- Direct recursion
  Explanation: Function A → Function B → Function A.

12. What is the main advantage of iteration over recursion?

Choices:

- **Iteration typically consumes less memory** ✅
- Iteration is easier to implement
- Iteration can handle larger input sizes
- Iteration is always faster
  Explanation: Loops do not use call stack.

13. In the context of algorithm analysis, what does the term "best case" refer to?

Choices:

- The maximum resources an algorithm might need
- The most common scenario encountered
- The average performance across all possible inputs
- **The minimum resources an algorithm might need** ✅
  Explanation: Best-case input → least operations.

14. What is the space complexity of an algorithm that uses a temporary array of size N?

Choices:

- O(log N)
- $O(N^2)$
- **O(N)** ✅
- O(1)
  Explanation: Memory grows linearly with input.

15. Which of the following represents the correct order of growth rates from smallest to largest?

Choices:

- **O(1), O(log N), O(N), O(N log N), $O(N^2)$** ✅
- O(log N), O(1), O(N), $O(N^2)$, O(N log N)
- O(1), O(N), O(log N), O(N log N), $O(N^2)$
- $O(N^2)$, O(N log N), O(N), O(log N), O(1)
  Explanation: Constant → logarithmic → linear → linearithmic → quadratic.

---

16. What is the time complexity of the following code snippet?

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

Choices:

- O(N²)
- O(N log N)
- O(log N)
- **O(N)** ✅
  Explanation: Executes N times → linear.

17. Which of the following is TRUE about Big Omega (Ω) notation?

Choices:

- It represents the exact running time
- It represents both upper and lower bounds
- It represents the upper bound of an algorithm's running time
- **It represents the lower bound of an algorithm's running time** ✅
  Explanation: Describes best-case/minimum performance.

18. What is the primary factor that determines the efficiency of an algorithm?

Choices:

- Both time and space complexity ✅
- Only space complexity
- The programming language used
- Only time complexity
  Explanation: Efficiency depends on speed and memory usage.
-

19. Which of the following scenarios is best suited for using recursion?

Choices:

- Problems involving simple arithmetic operations
- Problems requiring minimal memory usage
- **Problems that can be broken down into smaller, similar subproblems ✅**
- Problems where execution speed is critical
  Explanation: Works well with self-similar problem structure.

20. What is the output of the following recursive function when n=5?

```
void printNumbers(int n) { if(n == 0) return; System.out.println(n);
printNumbers(n-1); }
```

Choices:

- **5 4 3 2 1 ✅**
- 0 1 2 3 4
- 5 5 5 5 5
- 1 2 3 4 5
  Explanation: Prints n first, then calls itself with n-1.