

**1. Which of the following is a primitive data type in Java?**

- **Options:**
    - a) int
    - b) decimal
    - c) String
    - d) Object
  - **Correct Answer:** a) int
- 

**2. What is the size of the char data type in Java?**

- **Options:**
    - a) 8 bits
    - b) 16 bits
    - c) 32 bits
    - d) 64 bits
  - **Correct Answer:** b) 16 bits
- 

**3. Which data type is used to store a single-precision floating-point number in Java?**

- **Options:**
    - a) int
    - b) float
    - c) long
    - d) byte
  - **Correct Answer:** b) float
- 

**4. Which of the following is not a built-in data type in Java?**

- **Options:**
    - a) decimal
    - b) boolean
    - c) char
    - d) long
  - **Correct Answer:** a) decimal
- 

**5. What is the maximum value that can be stored in a byte variable in Java?**

- **Options:**
  - a) 128
  - b) 127
  - c) 256
  - d) 255

- **Correct Answer:** b) 127
- 

**6. What is the size of the double data type in Java?**

- **Options:**
    - a) 32 bits
    - b) 64 bits
    - c) 128 bits
    - d) 256 bits
  - **Correct Answer:** b) 64 bits
- 

**7. In Java, what is the range of values that can be stored in a short variable?**

- **Options:**
    - a) -128 to 127
    - b) -32768 to 32768
    - c) -32,768 to 32,767
    - d) -2,147,483,648 to 2,147,483,647
  - **Correct Answer:** c) -32,768 to 32,767
- 

**8. Which data type is used to represent whole numbers in Java?**

- **Options:**
    - a) float
    - b) int
    - c) double
    - d) byte
  - **Correct Answer:** b) int
- 

**9. Which data type is used to represent true/false values in Java?**

- **Options:**
    - a) boolean
    - b) int
    - c) char
    - d) byte
  - **Correct Answer:** a) boolean
- 

**10. What is the size of the boolean data type in Java?**

- **Options:**
  - a) 1 bit

- b) 8 bits
- c) 16 bits
- d) 32 bits

- **Correct Answer:** a) 1 bit

**1. What is the purpose of the System.out.print method in Java?**

- **Options:**
    - a) Reading user input
    - b) Writing text to the console
    - c) Opening a file
    - d) Performing arithmetic operations
  - **Correct Answer:** b) Writing text to the console
- 

**2. Which class provides methods for reading input from the user in Java?**

- **Options:**
    - a) Scanner
    - b) System
    - c) InputReader
    - d) BufferedReader
  - **Correct Answer:** a) Scanner
- 

**3. What is the method used to read a string from the user using Scanner?**

- **Options:**
    - a) readLine()
    - b) readString()
    - c) nextLine()
    - d) getString()
  - **Correct Answer:** c) nextLine()
- 

**4. How can you output a new line character in Java?**

- **Options:**
    - a) \n
    - b) \r
    - c) \t
    - d) \\
  - **Correct Answer:** a) \n
- 

**5. Which method is used to print the formatted output in Java?**

- **Options:**
    - a) print()
    - b) println()
    - c) printf()
    - d) write()
  - **Correct Answer:** c) printf()
- 

**6. What is the purpose of %n in a printf format string?**

- **Options:**
    - a) Represents a new line character
    - b) Represents a tab character
    - c) Represents the percentage symbol
    - d) Represents a null character
  - **Correct Answer:** a) Represents a new line character
- 

**7. Which operator is used to get the remainder of a division operation in Java?**

- **Options:**
    - a) %
    - b) &
    - c) /
    - d) \*
  - **Correct Answer:** a) %
- 

**8. What does the System.in represent in Java?**

- **Options:**
    - a) Standard output stream
    - b) Standard error stream
    - c) Standard input stream
    - d) Standard logging stream
  - **Correct Answer:** c) Standard input stream
- 

**9. What is the output of the following code?**

java

Copy code

```
int num = 10;
```

```
System.out.printf("The number is: %d", num);
```

- **Options:**
    - a) The number is: 10
    - b) The number is: "num"
    - c) The number is: %d
    - d) The number is: The number is: 10
  - **Correct Answer:** a) The number is: 10
- 

**10. What is the output of the following code?**

java

Copy code

```
double value = 12.345;
```

```
System.out.printf("Value: %.1f", value);
```

- **Options:**
  - a) Value: 12.3
  - b) Value: 12.345
  - c) Value: 12.34
  - d) Value: 12.35
- **Correct Answer:** d) Value: 12.35

**1. Which control statement is used to execute a block of code repeatedly based on a given condition?**

- **Options:**
    - a) if statement
    - b) for loop
    - c) while loop
    - d) do-while loop
  - **Correct Answer:** c) while loop
- 

**2. The switch statement in Java can only be used with which data types?**

- **Options:**
    - a) int and char
    - b) int and float
    - c) char and float
    - d) int and double
  - **Correct Answer:** a) int and char
- 

**3. What will be the output of the following code?**

java

Copy code

```
int x = 5;

if (x > 3) {

    System.out.println("x is greater than 3");

} else {

    System.out.println("x is not greater than 3");

}
```

- **Options:**
    - a) x is greater than 3
    - b) x is not greater than 3
    - c) x
    - d) 5
  - **Correct Answer:** a) x is greater than 3
- 

4. Which control statement is used to exit a loop prematurely when a specific condition is met?

- **Options:**
    - a) break statement
    - b) continue statement
    - c) return statement
    - d) exit statement
  - **Correct Answer:** a) break statement
- 

5. In a for loop, what happens if the update statement is omitted?

- **Options:**
    - a) The loop will not execute.
    - b) An error will occur during compilation.
    - c) The loop will run infinitely.
    - d) The loop will execute only once.
  - **Correct Answer:** c) The loop will run infinitely.
- 

6. What is the correct syntax for a switch statement in Java?

- **Options:**
  - a) switch(condition) { }
  - b) switch { }
  - c) case(condition) { }
  - d) case { }
- **Correct Answer:** a) switch(condition) { }

---

7. Which loop executes the loop body at least once, even if the condition is false?

- **Options:**
    - a) for loop
    - b) while loop
    - c) do-while loop
    - d) until loop
  - **Correct Answer:** c) do-while loop
- 

8. What is the purpose of the else statement in an if-else structure?

- **Options:**
    - a) To handle exceptions
    - b) To provide an alternative code block if the condition is false
    - c) To stop the execution of the program
    - d) To restart the execution of the program
  - **Correct Answer:** b) To provide an alternative code block if the condition is false
- 

9. What will be the output of the following code?

java

Copy code

```
int x = 10;
if (x % 2 == 0) {
    if (x > 5) {
        System.out.println("x is even and greater than 5");
    } else {
        System.out.println("x is even and not greater than 5");
    }
} else {
    System.out.println("x is odd");
}
```

- **Options:**
  - a) x is even and greater than 5
  - b) x is even and not greater than 5
  - c) x is odd
  - d) No output will be printed

- **Correct Answer:** a) x is even and greater than 5
- 

#### 10. What is the purpose of the break statement in a loop?

- **Options:**
  - a) To restart the loop
  - b) To skip the current iteration and move to the next one
  - c) To force the immediate termination of the loop
  - d) To return a value from the loop
- **Correct Answer:** c) To force the immediate termination of the loop

#### 1. What is an algorithm?

- **Options:**
    - a) A set of rules to decorate code
    - b) A step-by-step procedure to solve a problem
    - c) A programming language
    - d) A type of data structure
  - **Correct Answer:** b) A step-by-step procedure to solve a problem
- 

#### 2. Why do we need algorithms?

- **Options:**
    - a) To make code more complicated
    - b) To optimize resource usage and reduce execution time
    - c) To confuse other programmers
    - d) To avoid using data structures
  - **Correct Answer:** b) To optimize resource usage and reduce execution time
- 

#### 3. What characteristic ensures that an algorithm will eventually stop after a finite number of steps?

- **Options:**
    - a) Input
    - b) Output
    - c) Definiteness
    - d) Finiteness
  - **Correct Answer:** d) Finiteness
- 

#### 4. Which characteristic ensures that the algorithm's steps can be executed using basic operations or actions?

- **Options:**
  - a) Effectiveness
  - b) Correctness



- c) Determinism
- d) Feasibility

- **Correct Answer:** a) Effectiveness
- 

**5. What is a critical aspect of algorithm analysis?**

- **Options:**
    - a) Memory complexity
    - b) Syntax correctness
    - c) Runtime error handling
    - d) Proper indentation
  - **Correct Answer:** a) Memory complexity
- 

**6. Which type of algorithm aims to find a global optimum by making locally optimal choices?**

- **Options:**
    - a) Sorting Algorithms
    - b) Searching Algorithms
    - c) Greedy Algorithms
    - d) Backtracking Algorithms
  - **Correct Answer:** c) Greedy Algorithms
- 

**7. Which algorithmic technique breaks a problem into smaller sub-problems and solves them independently?**

- **Options:**
    - a) Dynamic Programming Algorithms
    - b) Divide and Conquer Algorithms
    - c) Backtracking Algorithms
    - d) Greedy Algorithms
  - **Correct Answer:** b) Divide and Conquer Algorithms
- 

**8. What ensures that an algorithm's behavior is well-defined and unambiguous?**

- **Options:**
    - a) Input
    - b) Output
    - c) Definiteness
    - d) Finiteness
  - **Correct Answer:** c) Definiteness
-

**9. Which type of algorithm is used to find a particular element in a data structure efficiently?**

- **Options:**
    - a) Sorting Algorithms
    - b) Graph Algorithms
    - c) Dynamic Programming Algorithms
    - d) Searching Algorithms
  - **Correct Answer:** d) Searching Algorithms
- 

**10. What is the primary purpose of standardizing algorithms?**

- **Options:**
  - a) To confuse other programmers
  - b) To make code more complicated
  - c) To improve code readability and consistency
  - d) To avoid using data structures
- **Correct Answer:** c) To improve code readability and consistency

**1. The space complexity of a recursive function is proportional to:**

- **Options:**
    - a) The number of iterations
    - b) The number of function calls
    - c) The number of conditional statements
    - d) The number of variables used
  - **Correct Answer:** b) The number of function calls
- 

**2. To verify whether a function grows faster or slower than the other function, we have some asymptotic or mathematical notations, which is\_\_\_\_\_.**

- **Options:**
    - a) Big Omega  $\Omega(f)$
    - b) Big Theta  $\Theta(f)$
    - c) Big Oh  $O(f)$
    - d) All of the above
  - **Correct Answer:** d) All of the above
- 

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

- **Options:**
  - a) X will always be a better choice for small inputs
  - b) Y will always be a better choice for small inputs
  - c) X will always be a better choice for large inputs
  - d) X will always be a better choice for all inputs

- **Correct Answer:** c) X will always be a better choice for large inputs
- 

4. Which of the following is an example of an  $O(1)$  space complexity algorithm?

- **Options:**
    - a) Quicksort
    - b) Merge sort
    - c) Bubble sort
    - d) Heap sort
  - **Correct Answer:** c) Bubble sort
- 

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

- **Options:**
    - a) describes limiting behavior of the function
    - b) characterizes a function based on growth of function
    - c) upper bound on growth rate of the function
    - d) all of the mentioned
  - **Correct Answer:** d) all of the mentioned
- 

6. Which of the following is the most efficient algorithm in terms of time complexity?

- **Options:**
    - a)  $O(n)$
    - b)  $O(n^2)$
    - c)  $O(n \log n)$
    - d)  $O(2^n)$
  - **Correct Answer:** a)  $O(n)$
- 

7. What is the time complexity of an algorithm that performs  $n/2$  operations in the worst case scenario?

- **Options:**
    - a)  $O(n)$
    - b)  $O(\log n)$
    - c)  $O(n \log n)$
    - d)  $O(n^2)$
  - **Correct Answer:** a)  $O(n)$
- 

8.  $O(\log n)$  is?

- **Options:**
    - a) constant asymptotic notations
    - b) logarithmic asymptotic notations
    - c) polynomial asymptotic notations
    - d) quadratic asymptotic notations
  - **Correct Answer:** b) logarithmic asymptotic notations
- 

**9. What is the time complexity of the following code:**

c

Copy code

```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

- **Options:**
    - a)  $O(N)$
    - b)  $O(N \log(N))$
    - c)  $O(N * \text{Sqrt}(N))$
    - d)  $O(NN)$
  - **Correct Answer:** d)  $O(N*N)$
- 

**10. What is the space complexity of an algorithm that uses an array of size n to store the input, and another array of size m to store the output?**

- **Options:**
  - a)  $O(1)$
  - b)  $O(n)$
  - c)  $O(m)$
  - d)  $O(n + m)$
- **Correct Answer:** d)  $O(n + m)$

**1. Which of the following is an optimization used in the Sieve of Eratosthenes algorithm to reduce memory usage?**

- **Options:**
  - A. Using a linked list instead of an array
  - B. Marking only odd numbers as prime

- C. Using a heap instead of a stack
- D. None of the above

- **Correct Answer:** B. Marking only odd numbers as prime
- 

## 2. What is the space complexity of Simple Sieve?

- **Options:**
    - A.  $O(1)$
    - B.  $O(n)$
    - C.  $O(n \log n)$
    - D.  $O(n^2)$
  - **Correct Answer:** B.  $O(n)$
- 

## 3. Which of the following is not a step in the Sieve of Eratosthenes algorithm?

- **Options:**
    - A. Create a boolean array of size  $n+1$
    - B. Mark all multiples of 2 as composite
    - C. Mark all multiples of 3 as composite
    - D. Find the greatest common divisor of each number
  - **Correct Answer:** D. Find the greatest common divisor of each number
- 

## 4. What is the condition used to mark a number as composite in Simple Sieve?

- **Options:**
    - A. The number is divisible by 2
    - B. The number is divisible by 3
    - C. The number is divisible by any prime less than its square root
    - D. The number is divisible by any prime less than itself
  - **Correct Answer:** C. The number is divisible by any prime less than its square root
- 

## 5. How many prime numbers less than or equal to 100 can be found using the Sieve of Eratosthenes algorithm?

- **Options:**
    - A. 24
    - B. 25
    - C. 26
    - D. 13
  - **Correct Answer:** B. 25
-

**6. Which of the following optimizations can be applied to Simple Sieve to reduce its memory usage?**

- **Options:**
    - A. Storing only odd numbers in the sieve
    - B. Using bit packing to store the boolean flags
    - C. Both A and B
    - D. None of the above
  - **Correct Answer:** C. Both A and B
- 

**7. Which of the following is a drawback of the Simple Sieve algorithm?**

- **Options:**
    - A. Requires less memory
    - B. Requires more memory
    - C. Takes less time to execute
    - D. Takes more time to execute
  - **Correct Answer:** B. Requires more memory
- 

**8. Which of the following data structures can be used to implement Simple Sieve efficiently?**

- **Options:**
    - A. Array
    - B. Linked list
    - C. Stack
    - D. Queue
  - **Correct Answer:** A. Array
- 

**9. Which of the following algorithms is an improvement over Simple Sieve?**

- **Options:**
    - A. Segmented Sieve
    - B. Incremental Sieve
    - C. Both A and B
    - D. None of the above
  - **Correct Answer:** C. Both A and B
- 

**10. How many prime numbers are there between 100 and 200?**

- **Options:**
  - A. 23
  - B. 21

- C. 10
- D. 12

- **Correct Answer:** A. 23

**1. What is the main advantage of the segmented sieve algorithm over the traditional sieve of Eratosthenes algorithm?**

- **Options:**
    - a) It uses less memory.
    - b) It is faster for small ranges of numbers.
    - c) It is easier to implement.
    - d) It can handle larger ranges of numbers.
  - **Correct Answer:** d) It can handle larger ranges of numbers
- 

**2. Which of the following is true about the incremental sieve?**

- **Options:**
    - a) It only finds prime numbers.
    - b) It finds both prime and composite numbers.
    - c) It only finds composite numbers.
    - d) It does not find any numbers.
  - **Correct Answer:** a) It only finds prime numbers
- 

**3. What is the output of the incremental sieve when n=20 is given as input?**

- **Options:**
    - a) 2 3 5 7 11 13 17
    - b) 2 3 5 7 11 13 17 19 20
    - c) 2 3 5 7 11 13 17 19
    - d) 2 3 5 7 12 13 17 19
  - **Correct Answer:** c) 2 3 5 7 11 13 17 19
- 

**4. What is the time complexity of the given code snippet?**

python

Copy code

f():

```
ans = 0
```

```
for i = 1 to n:
```

```
    for j = i; j <= n; j += i:
```

```
        ans += 1
```

```
print(ans)
```

- **Options:**
    - a)  $O(\log n)$
    - b)  $O(n)$
    - c)  $O(n \log n)$
    - d)  $O(n * n)$
  - **Correct Answer:** c)  $O(n \log n)$
- 

5. What is the space complexity of the segmented sieve?

- **Options:**
    - a)  $O(n)$
    - b)  $O(n \log n)$
    - c)  $O(n \log \log n)$
    - d)  $O(\sqrt{n})$
  - **Correct Answer:** d)  $O(\sqrt{n})$
- 

6. Which of the following is a disadvantage of the incremental sieve?

- **Options:**
    - a) It requires a large amount of memory.
    - b) It is slower than the segmented sieve.
    - c) It only works for small ranges.
    - d) It can only find prime numbers.
  - **Correct Answer:** b) It is slower than the segmented sieve
- 

7. Which of the following is a disadvantage of the segmented sieve?

- **Options:**
    - a) It requires a large amount of memory.
    - b) It is slower than the incremental sieve.
    - c) It only works for small ranges.
    - d) It can only find prime numbers.
  - **Correct Answer:** a) It requires a large amount of memory
- 

8. What is the formula for calculating the nth prime number using the Incremental Sieve algorithm?

- **Options:**
  - a)  $n * (n - 1) + 2$
  - b)  $n * n - n + 1$



c)  $n * n + n + 1$

d)  $n * n + 2$

- **Correct Answer:** d)  $n * n + 2$
- 

**9. Which algorithm is best suited for finding all prime numbers in a small range?**

- **Options:**
  - a) Segmented sieve
  - b) Incremental sieve
  - c) Both algorithms are equally suited for this task.
  - d) Neither algorithm is suited for this task.

- **Correct Answer:** b) Incremental sieve
- 

**10. What is the advantage of using the Segmented Sieve algorithm over the trial division method for prime number generation?**

- **Options:**
  - a) The Segmented Sieve algorithm is faster
  - b) The Segmented Sieve algorithm is simpler to implement
  - c) The Segmented Sieve algorithm can be parallelized
  - d) The Segmented Sieve algorithm can generate prime numbers up to a larger limit

- **Correct Answer:** a) The Segmented Sieve algorithm is faster

**1. What is Euler's phi function?**

- **Options:**
  - A) A function that counts the number of divisors of an integer.
  - B) A function that counts the number of prime factors of an integer.
  - C) A function that counts the number of integers less than a given integer  $n$  that are relatively prime to  $n$ .
  - D) A function that counts the number of integers less than a given integer  $n$  that are divisible by  $n$ .

- **Correct Answer:** C) A function that counts the number of integers less than a given integer  $n$  that are relatively prime to  $n$ .
- 

**2. What is the relationship between the values of  $\varphi(n)$  and  $\varphi(p^k)$  for prime  $p$  and positive integer  $k$ ?**

- **Options:**
  - A)  $\varphi(n) = p^k - 1$
  - B)  $\varphi(n) = p^{(k-1)}$
  - C)  $\varphi(n) = p^k$
  - D)  $\varphi(n) = (p-1) * p^{(k-1)}$

- **Correct Answer:** D)  $\varphi(n) = (p-1) * p^{(k-1)}$

---

3. What is the relationship between the values of  $\varphi(n)$  and  $\varphi(m)$  for coprime positive integers  $n$  and  $m$ ?

- **Options:**
    - A)  $\varphi(nm) = \varphi(n) + \varphi(m)$
    - B)  $\varphi(nm) = \varphi(n)\varphi(m)$
    - C)  $\varphi(nm) = \varphi(n) - \varphi(m)$
    - D) None of the above
  - **Correct Answer:** B)  $\varphi(nm) = \varphi(n)\varphi(m)$
- 

4. What is the output of  $\phi(324)$ ?

- **Options:**
    - A) 98
    - B) 90
    - C) 108
    - D) 120
  - **Correct Answer:** B) 90
- 

5. What is the  $\phi$  function of the number 3?

- **Options:**
    - A) 0
    - B) 1
    - C) 2
    - D) 3
  - **Correct Answer:** C) 2
- 

6. What is the  $\phi$  function of the number 8?

- **Options:**
    - A) 0
    - B) 1
    - C) 2
    - D) 4
  - **Correct Answer:** D) 4
- 

7. What is the value of  $\phi(n)$  for  $n = 12$  using Euler's  $\phi$  function?

- **Options:**
  - A) 1
  - B) 2

- C) 6
- D) 4

- **Correct Answer:** D) 4
- 

**8. What is the  $\phi$  function of the number 12?**

- **Options:**
  - A) 0
  - B) 1
  - C) 2
  - D) 4

- **Correct Answer:** D) 4
- 

**9. Which of the following is a property of Euler's phi function?**

- **Options:**
  - A)  $\phi(n)$  is always even for any positive integer  $n$ .
  - B)  $\phi(n)$  is equal to the number of divisors of  $n$ .
  - C)  $\phi(p) = p-1$  for any prime number  $p$ .
  - D)  $\phi(n)$  is always greater than  $n$  for any positive integer  $n$ .

- **Correct Answer:** C)  $\phi(p) = p-1$  for any prime number  $p$ .
- 

**10. What is the value of Euler's Totient Function for the number 2000?**

- **Options:**
  - A) 789
  - B) 880
  - C) 800
  - D) 670

- **Correct Answer:** B) 880

**1. What is a Strobogrammatic Number?**

- **Options:**
  - A) A number that is divisible by 3
  - B) A number that reads the same when rotated 180 degrees
  - C) A number that is a multiple of 5
  - D) A number that contains only even digits

- **Correct Answer:** B) A number that reads the same when rotated 180 degrees
- 

**2. What is the strobogrammatic number that is the largest prime number?**

- **Options:**  
A) 11  
B) 101  
C) 181  
D) There is no strobogrammatic number that is prime
  - **Correct Answer:** B) 101
- 

**3. Which of the following is not a Strobogrammatic Number?**

- **Options:**  
A) 818  
B) 1881  
C) 969  
D) 666
  - **Correct Answer:** D) 666
- 

**4. Which of the following statements is true about Strobogrammatic Numbers?**

- **Options:**  
A) They are only found in base 10  
B) They are always palindromic  
C) They can be used in cryptography to encode messages  
D) They are always even
  - **Correct Answer:** A) They are only found in base 10
- 

**5. A number that reads the same when rotated 180 degrees. What is the strobogrammatic number that is the largest prime number?**

- **Options:**  
A) 11  
B) 101  
C) 181  
D) There is no strobogrammatic number that is prime
  - **Correct Answer:** B) 101
- 

**6. Which of the following is not a Strobogrammatic Number?**

- **Options:**  
A) 818  
B) 1881  
C) 969  
D) 666

- **Correct Answer:** D) 666
- 

**7. Which of the following statements is true about Strobogrammatic Numbers?**

- **Options:**
    - A) They are only found in base 10
    - B) They are always palindromic
    - C) They can be used in cryptography to encode messages
    - D) They are always even
  - **Correct Answer:** A) They are only found in base 10
- 

**8. Which of the following is a property of strobogrammatic numbers?**

- **Options:**
    - A) They are always prime
    - B) They are always even
    - C) They are always palindromic
    - D) They are always odd
  - **Correct Answer:** C) They are always palindromic
- 

**9. What is an advantage of using Strobogrammatic numbers in cryptography?**

- **Options:**
    - A) They provide a higher level of security.
    - B) They enable faster encryption and decryption processes.
    - C) They are resistant to data breaches and hacking.
    - D) They improve the efficiency of key generation.
  - **Correct Answer:** A) They provide a higher level of security.
- 

**10. Which of the following is a disadvantage of Strobogrammatic numbers in arithmetic operations?**

- **Options:**
    - A) They lead to inaccurate results in division operations.
    - B) They require specialized algorithms for addition and subtraction.
    - C) They cause a slowdown in multiplication and exponentiation.
    - D) They are incompatible with decimal and fractional numbers.
  - **Correct Answer:** B) They require specialized algorithms for addition and subtraction.
- 

**1. Find  $x$  such that  $3x \equiv 7 \pmod{10}$**

- **Options:**  
A)  $x \equiv 19 \pmod{10}$   
B)  $x \equiv 9 \pmod{11}$   
C)  $x \equiv 7 \pmod{10}$   
D)  $x \equiv 9 \pmod{10}$
  - **Correct Answer:** D)  $x \equiv 9 \pmod{10}$
- 

**2. Which of the following statements is true regarding the Chinese Remainder Theorem?**

- **Options:**  
A) It can only be applied to solve systems of congruences with prime moduli.  
B) It can be applied to solve systems of congruences with coprime moduli.  
C) It can be applied to solve systems of congruences with any moduli.  
D) It can only be applied to solve systems of congruences with integer moduli.
  - **Correct Answer:** B) It can be applied to solve systems of congruences with coprime moduli.
- 

**3. What will be the output of the following code?**

java

Copy code

```
int num = 8;  
int divisor = 3;  
int quotient = num / divisor;  
int remainder = num % divisor;  
System.out.println(quotient + " " + remainder);
```

- **Options:**  
A) 3 3  
B) 2 2  
C) 1 2  
D) 3 4
  - **Correct Answer:** C) 1 2
- 

**4. Consider a number that leaves a remainder of 2 when divided by 3, a remainder of 4 when divided by 5, and a remainder of 6 when divided by 7. What is the number according to the Chinese Remainder Theorem?**

- **Options:**  
A) 114  
B) 119

C) 123

D) 456

- **Correct Answer:** B) 119
- 

#### 5. What is the remainder theorem?

- **Options:**
    - A) It is a theorem in calculus that relates to finding the remainder of a polynomial function after dividing it by another polynomial function.
    - B) It is a theorem in number theory that states that for any integer  $n$  and any integer  $a$ , there exists a unique integer  $q$  and  $r$  such that  $n = aq + r$ , where  $r$  is the remainder.
    - C) It is a theorem in algebra that states that if a polynomial  $f(x)$  is divided by  $x-a$ , then the remainder is equal to  $f(a)$ .
    - D) It is a theorem in statistics that relates to finding the remainder of a set of data after dividing it by another set of data.
  - **Correct Answer:** B) It is a theorem in number theory that states that for any integer  $n$  and any integer  $a$ , there exists a unique integer  $q$  and  $r$  such that  $n = aq + r$ , where  $r$  is the remainder.
- 

#### 6. The Chinese Remainder Theorem is often used in which field?

- **Options:**
    - A) Number theory
    - B) Geometry
    - C) Algebra
    - D) Calculus
  - **Correct Answer:** A) Number theory
- 

#### 7. Which of the following is a disadvantage of the Chinese Remainder Theorem?

- **Options:**
    - A) It requires advanced knowledge of complex numbers.
    - B) It can only be applied to linear equations.
    - C) It has limited applicability to certain types of problems.
    - D) It is computationally intensive and time-consuming.
  - **Correct Answer:** C) It has limited applicability to certain types of problems.
- 

#### 8. Consider a number that leaves a remainder of 2 when divided by 5, a remainder of 3 when divided by 7, and a remainder of 4 when divided by 9. What is the number according to the Chinese Remainder Theorem?

- **Options:**
  - A) 76

- B) 158
- C) 156
- D) 67

- **Correct Answer:** C) 156
- 

9.  $x \equiv 1 \pmod{5}$ ,  $x \equiv 3 \pmod{7}$ . What is the smallest positive integer that satisfies these congruences using the Chinese Remainder Theorem?

- **Options:**

- A) 17
- B) 11
- C) 10
- D) 13

- **Correct Answer:** D) 13
- 

10. In cryptography, the Chinese Remainder Theorem is used for:

- **Options:**

- A) Generating random numbers.
- B) Encrypting messages.
- C) Decrypting messages.
- D) Hashing algorithms.

- **Correct Answer:** B) Encrypting messages.

1. What is the shape of an hourglass in a 2D array?

- **Options:**

- A) Square
- B) L-shape
- C) T-shape
- D) Hourglass

- **Correct Answer:** D) Hourglass
- 

2. How many elements does a standard hourglass in a 2D array contain?

- **Options:**

- A) 5
- B) 6
- C) 7
- D) 8

- **Correct Answer:** C) 7
- 

3. In a 6x6 matrix, how many distinct hourglasses can be found?



- **Options:**
    - A) 16
    - B) 24
    - C) 36
    - D) 49
  - **Correct Answer:** B) 24
- 

**4. Which of the following best describes the process of finding an hourglass sum in a matrix?**

- **Options:**
    - A) Calculate the sum of all elements in the matrix.
    - B) Calculate the sum of elements in each possible hourglass shape.
    - C) Calculate the sum of elements in each row.
    - D) Calculate the sum of elements in each column.
  - **Correct Answer:** B) Calculate the sum of elements in each possible hourglass shape.
- 

**5. What is the time complexity of finding the maximum hourglass sum in a 6x6 matrix?**

- **Options:**
    - A)  $O(1)$
    - B)  $O(n)$
    - C)  $O(n^2)$
    - D)  $O(n^3)$
  - **Correct Answer:** C)  $O(n^2)$
- 

**6. Which of the following loops is essential in the implementation of finding an hourglass sum?**

- **Options:**
    - A) A single loop iterating over all elements.
    - B) Two nested loops iterating over rows and columns.
    - C) Three nested loops iterating over rows, columns, and diagonals.
    - D) A loop iterating only over diagonals.
  - **Correct Answer:** B) Two nested loops iterating over rows and columns.
- 

**7. What should be initialized before iterating through the matrix to find the maximum hourglass sum?**

- **Options:**
  - A) Minimum possible integer value
  - B) Maximum possible integer value
  - C) Zero
  - D) Matrix size

- **Correct Answer:** A) Minimum possible integer value
- 

**8. Which of the following Java methods is most suitable for storing hourglass sums while iterating through the matrix?**

- **Options:**
    - A) ArrayList
    - B) Linked List
    - C) Priority Queue
    - D) int variable
  - **Correct Answer:** D) int variable
- 

**9. Which part of the hourglass is typically ignored when calculating its sum?**

- **Options:**
    - A) Top row
    - B) Bottom row
    - C) Center element
    - D) None
  - **Correct Answer:** D) None
- 

**10. How is the maximum hourglass sum finally determined?**

- **Options:**
  - A) By summing all hourglass sums.
  - B) By finding the minimum hourglass sum.
  - C) By comparing each hourglass sum with the current maximum.
  - D) By counting the total number of hourglasses.
- **Correct Answer:** C) By comparing each hourglass sum with the current maximum.

**1. What is the time complexity of the max() method in a Priority Queue in Java?**

- **Options:**
    - A)  $O(1)$
    - B)  $O(\log N)$
    - C)  $O(N)$
    - D)  $O(N \log N)$
  - **Correct Answer:** C)  $O(N)$
- 

**2. In a concurrent environment, which of the following Java classes is best suited for maintaining a maximum value with safe thread operations?**

- **Options:**
  - A) ConcurrentSkipListMap

- B) ConcurrentHashMap
- C) SynchronizedList
- D) CopyOnWriteArrayList

- **Correct Answer:** A) ConcurrentSkipListMap
- 

**3. If you have a TreeMap in Java and you want to find the maximum key, which method would you use?**

- **Options:**
    - A) lastKey()
    - B) floorKey()
    - C) higherKey()
    - D) firstKey()
  - **Correct Answer:** A) lastKey()
- 

**4. What is the primary reason to use a TreeSet when needing to maintain a maximum equilibrium in a sorted collection?**

- **Options:**
    - A) It maintains natural ordering and does not allow duplicates.
    - B) It is synchronized and thread-safe.
    - C) It offers constant-time performance for basic operations.
    - D) It allows null values for maximum flexibility.
  - **Correct Answer:** A) It maintains natural ordering and does not allow duplicates.
- 

**5. How does the PriorityQueue class in Java determine the maximum element in a priority queue?**

- **Options:**
    - A) By using a max heap internally.
    - B) By using a min heap internally.
    - C) By using a balanced binary tree.
    - D) By sorting elements at runtime.
  - **Correct Answer:** B) By using a min heap internally.
- 

**6. What is the maximum size of an array in Java on a 32-bit JVM?**

- **Options:**
  - A)  $2^{16} - 1$
  - B)  $2^{31} - 1$
  - C)  $2^{24} - 1$
  - D)  $2^{30} - 1$
- **Correct Answer:** B)  $2^{31} - 1$

---

**7. Which of the following is the most efficient way to keep track of the maximum value in a dynamic array when inserting and deleting elements frequently?**

- **Options:**
    - A) Use a balanced binary search tree.
    - B) Use a max-heap data structure.
    - C) Use an ArrayList with sorting after each insertion.
    - D) Use a HashMap to track maximum values.
  - **Correct Answer:** B) Use a max-heap data structure.
- 

**8. When using Collections.max() method, what is the time complexity if the collection is a List?**

- **Options:**
    - A)  $O(1)$
    - B)  $O(\log N)$
    - C)  $O(N)$
    - D)  $O(N \log N)$
  - **Correct Answer:** C)  $O(N)$
- 

**9. In Java, which of the following classes implements the Navigable Map interface and supports operations for retrieving the maximum key?**

- **Options:**
    - A) HashMap
    - B) TreeMap
    - C) LinkedHashMap
    - D) ConcurrentHashMap
  - **Correct Answer:** B) TreeMap
- 

**10. Given the following code snippet, what is the expected output if the PriorityQueue is used for storing integers?**

java

Copy code

```
PriorityQueue<Integer> pq = new PriorityQueue<>(Collections.reverseOrder());  
pq.add(10);  
pq.add(20);  
pq.add(30);  
System.out.println(pq.peek());
```

- **Options:**

- A) 10
- B) 20
- C) 30
- D) 40

- **Correct Answer:** C) 30

1. Which algorithm has the best time complexity for finding all leaders in an array?

- **Options:**

- A)  $O(n \log n)$  using sorting
- B)  $O(n)$  using a single traversal
- C)  $O(n^2)$  using nested loops
- D)  $O(n^2)$  using a priority queue

- **Correct Answer:** B)  $O(n)$  using a single traversal

---

2. Given an array A of size n, what is the maximum number of leaders possible in the array?

- **Options:**

- A) 1
- B)  $n-1$
- C)  $n$
- D)  $n/2$

- **Correct Answer:** B)  $n-1$

---

3. In an array with all elements being the same, how many leaders are there?

- **Options:**

- A) 0
- B) 1
- C)  $n-2$
- D)  $n$

- **Correct Answer:** B) 1

---

4. Consider the array A = [2, 4, 3, 6, 5, 8]. Which of the following elements is a leader?

- **Options:**

- A) 2
- B) 4
- C) 6
- D) 8

- **Correct Answer:** D) 8

---

5. If an array contains negative numbers only, what is true about the leader of the array?

- **Options:**
    - A) The smallest negative number is always a leader.
    - B) The largest negative number is always a leader.
    - C) No negative number can be a leader.
    - D) All negative numbers are leaders.
  - **Correct Answer:** B) The largest negative number is always a leader.
- 

6. Which of the following operations can help in finding leaders efficiently in a large array?

- **Options:**
    - A) Precomputing the maximum values for suffixes
    - B) Sorting the array and then scanning
    - C) Using a hash map to store all elements
    - D) Computing all possible subarrays
  - **Correct Answer:** A) Precomputing the maximum values for suffixes
- 

7. Given an array  $A = [1, 2, 3, 4, 5, 6]$ , which of the following statements is true about leaders in this array?

- **Options:**
    - A) All elements are leaders.
    - B) Only the last element is a leader.
    - C) The first element is a leader.
    - D) No elements are leaders.
  - **Correct Answer:** B) Only the last element is a leader.
- 

8. For an array  $A$  of length  $n$ , what is the space complexity of the optimal solution for finding all leaders?

- **Options:**
    - A)  $O(n)$
    - B)  $O(\log n)$
    - C)  $O(1)$
    - D)  $O(n^2)$
  - **Correct Answer:** A)  $O(n)$
- 

9. Which data structure can be utilized to efficiently keep track of the maximum elements seen so far while traversing an array from right to left?

- **Options:**
  - A) Stack

- B) Queue
- C) Linked List
- D) Binary Search Tree

- **Correct Answer:** A) Stack
- 

10. In an array A where A = [5, 7, 3, 7, 6, 8], which of the following pairs of elements could be leaders if we choose the array A to be a modified version where some elements are shifted?

- **Options:**
  - A) 5 and 8
  - B) 3 and 6
  - C) 7 and 8
  - D) 7 and 6
- **Correct Answer:** C) 7 and 8

1. Which of the following algorithms is commonly used to find the majority element in an array in  $O(n)$  time and  $O(1)$  space?

- **Options:**
    - A) QuickSort
    - B) MergeSort
    - C) Boyer-Moore Voting Algorithm
    - D) Binary Search
  - **Correct Answer:** C) Boyer-Moore Voting Algorithm
- 

2. What is the time complexity of the Boyer-Moore Voting Algorithm for finding the majority element?

- **Options:**
    - A)  $O(n)$
    - B)  $O(n \log n)$
    - C)  $O(n^2)$
    - D)  $O(\log n)$
  - **Correct Answer:** A)  $O(n)$
- 

3. Which of the following conditions must be true for an element to be considered a majority element in an array of size  $n$ ?

- **Options:**

- ☐ A) It appears at least  $n/4$  times.
  - ☐ B) It appears at least  $n/3$  times.
  - ☐ C) It appears at least  $n/2$  times.
  - ☐ D) It appears at least  $n$  times.
- **Correct Answer:** C) It appears at least  $n/2$  times.
- 

4. Given the following Java code, what will be the output if the input array is [1, 2, 3, 2, 2, 1, 2, 2]?

java

Copy code

```
public int majorityElement(int[] nums) {  
    int count = 0, candidate = 0;  
    for (int num : nums) {  
        if (count == 0) {  
            candidate = num;  
        }  
        count += (num == candidate) ? 1 : -1;  
    }  
    return candidate;  
}
```

- **Options:**
  - ☐ A) 1
  - ☐ B) 2
  - ☐ C) 3
  - ☐ D) 0

- **Correct Answer:** B) 2
- 

5. Which of the following Java code snippets correctly initializes the candidate and count variables for the Boyer-Moore Voting Algorithm?

- **Options:**
  - ☐ A) int candidate = 0, count = 1;
  - ☐ B) int candidate = nums[0], count = 0;



- ☐ C) int candidate = -1, count = 0;
  - ☐ D) int candidate = 0, count = 0;
  - **Correct Answer:** D) int candidate = 0, count = 0;
- 

**6. In the Boyer-Moore Voting Algorithm, what happens when the count variable becomes zero?**

- **Options:**
    - ☐ A) The candidate is confirmed as the majority element.
    - ☐ B) The candidate is discarded, and a new candidate is chosen.
    - ☐ C) The algorithm terminates.
    - ☐ D) The count is reset to 1.
  - **Correct Answer:** B) The candidate is discarded, and a new candidate is chosen.
- 

**7. Given the following Java code, what will be the output if the input array is [2, 2, 1, 1, 1, 2, 2]?**

java

Copy code

```
public int majorityElement(int[] nums) {  
    int count = 0, candidate = 0;  
    for (int num : nums) {  
        if (count == 0) {  
            candidate = num;  
        }  
        count += (num == candidate) ? 1 : -1;  
    }  
    return candidate;  
}
```

- **Options:**
  - ☐ A) 1
  - ☐ B) 2
  - ☐ C) 0
  - ☐ D) -1
- **Correct Answer:** B) 2

---

8. Consider the following array: [3, 3, 4, 2, 4, 4, 2, 4, 4]. Which element is the majority element according to the Boyer-Moore Voting Algorithm?

- Options:
  - A) 2
  - B) 3
  - C) 4
  - D) No majority element
- Correct Answer: C) 4

---

9. Which Java method can be used to verify if a candidate is indeed the majority element after using the Boyer-Moore Voting Algorithm?

- Options:
  - A) Collections.frequency()
  - B) Arrays.binarySearch()
  - C) Stream.count()
  - D) Arrays.sort()
- Correct Answer: A) Collections.frequency()

---

10. If an array contains multiple majority elements, which of the following statements is true for the Boyer-Moore Voting Algorithm?

- Options:
  - A) It will return the first majority element it encounters.
  - B) It will return a random majority element.
  - C) It will fail to find any majority element.
  - D) There can only be one majority element.
- Correct Answer: D) There can only be one majority element.

1. What is the lexicographically first palindromic string among the following?

- Options:
  - A. madam
  - B. civic
  - C. aibohphobia

- D. radar
  - **Correct Answer:** A. madam  
*Explanation: "madam" comes first alphabetically when compared to the other palindromic strings.*
- 

## 2. Which method in Java is used to compare two strings lexicographically?

- **Options:**
    - A. equals()
    - B. compareTo()
    - C. charAt()
    - D. substring()
  - **Correct Answer:** B. compareTo()  
*Explanation: The compareTo() method compares two strings lexicographically based on the Unicode value of each character.*
- 

## 3. What is the output of the following Java code snippet?

java

Copy code

```
String str = "racecar";  
StringBuilder sb = new StringBuilder(str);  
if (str.equals(sb.reverse().toString())) {  
    System.out.println("Palindrome");  
} else {  
    System.out.println("Not Palindrome");  
}
```

- **Options:**
    - A. Palindrome
    - B. Not Palindrome
    - C. Compile Error
    - D. Runtime Error
  - **Correct Answer:** A. Palindrome  
*Explanation: The string "racecar" is a palindrome, so the output will be "Palindrome."*
-

4. Which of the following approaches can be used to check if a string is a palindrome?

- **Options:**

- ☐ A. Two-pointer technique
- ☐ B. Recursion
- ☐ C. Using a stack
- ☐ D. All of the above

- **Correct Answer:** D. All of the above

*Explanation: All listed techniques can be used to check if a string is a palindrome.*

---

5. Given the string `str = "aabbcc"`, what will be the lexicographically first palindromic string that can be formed?

- **Options:**

- ☐ A. abccba
- ☐ B. abcba
- ☐ C. accbca
- ☐ D. No palindrome can be formed

- **Correct Answer:** A. abccba

*Explanation: A valid palindrome can be formed by sorting the characters and forming the lexicographically first palindrome.*

---

6. Which of the following is the correct approach to find the lexicographically first palindromic string in Java?

- **Options:**

- ☐ A. Sort the string, then form a palindrome.
- ☐ B. Find all permutations of the string and check each for palindromes.
- ☐ C. Use a frequency array to count characters, then form the palindrome.
- ☐ D. Use dynamic programming to find the longest palindromic subsequence.

- **Correct Answer:** A. Sort the string, then form a palindrome.

*Explanation: Sorting the string allows the formation of the lexicographically first palindrome.*

---

7. What does the following Java code snippet print?

java

Copy code

```
String str = "aabb";  
char[] chars = str.toCharArray();  
Arrays.sort(chars);  
String sortedStr = new String(chars);  
System.out.println(sortedStr);
```

- **Options:**

- A. bbaa
- B. aabb
- C. abab
- D. baba

- **Correct Answer:** B. aabb

*Explanation: Sorting the characters in "aabb" results in the same string "aabb" since it's already sorted.*

---

**8. If a string contains an odd number of characters, which condition must be met to form a palindromic string?**

- **Options:**

- A. All characters must occur an even number of times.
- B. All characters except one must occur an even number of times.
- C. At least one character must occur an odd number of times.
- D. All characters must be distinct.

- **Correct Answer:** B. All characters except one must occur an even number of times.

*Explanation: A palindrome can have one character with an odd frequency (which will be in the center) and all other characters must have even frequencies.*

---

**9. What is the purpose of the reverse() method in the StringBuilder class?**

- **Options:**

- A. To reverse the characters in a string
- B. To compare two strings
- C. To find the lexicographically smallest string
- D. To split a string into substrings

- **Correct Answer:** A. To reverse the characters in a string

*Explanation: The reverse() method reverses the characters in a StringBuilder object.*

---

10. Which data structure is best suited to keep track of character frequencies when forming a palindromic string?

- **Options:**

- A. Stack
- B. Queue
- C. HashMap
- D. LinkedList

- **Correct Answer:** C. HashMap

*Explanation: A HashMap is ideal for storing character frequencies because it allows quick lookup and update of character counts.*

1. Which of the following classes in Java implements the Comparable interface to define natural ordering?

- **Options:**

- A. HashMap
- B. ArrayList
- C. String
- D. LinkedList

- **Correct Answer:** C. String

*Explanation: The String class implements the Comparable interface to define its natural ordering based on lexicographical order.*

---

2. What method must a class implement to define its natural order?

- **Options:**

- A. compareTo()
- B. compare()
- C. sort()
- D. order()

- **Correct Answer:** A. compareTo()

*Explanation: To define the natural order of a class, it must implement the compareTo() method from the Comparable interface.*

---

3. What will be the output of the following code?

java

Copy code

```
List<Integer> list = Arrays.asList(3, 2, 1);
```

```
Collections.sort(list);
```

```
System.out.println(list);
```

- **Options:**

- A. [1, 2, 3]
- B. [3, 2, 1]
- C. [2, 3, 1]
- D. [1, 3, 2]

- **Correct Answer:** A. [1, 2, 3]

*Explanation: The Collections.sort() method sorts the list in ascending order, so the output will be [1, 2, 3].*

---

**4. How does the compareTo method in the Comparable interface signal that one object is greater than, equal to, or less than another?**

- **Options:**

- A. By returning a boolean value
- B. By throwing an exception
- C. By returning an integer
- D. By calling another method

- **Correct Answer:** C. By returning an integer

*Explanation: The compareTo() method returns an integer: a negative value if the object is less than the other, zero if they are equal, and a positive value if the object is greater.*

---

**5. Which class in Java uses the natural ordering of its elements?**

- **Options:**

- A. HashSet
- B. TreeSet
- C. LinkedList
- D. Vector

- **Correct Answer:** B. TreeSet

*Explanation: The TreeSet class uses the natural ordering of its elements, or an ordering defined by a Comparator.*

---

6. Given the following class definition, what is true about the compareTo method?

java

Copy code

```
public class Person implements Comparable<Person> {  
    private String name;  
  
    @Override  
    public int compareTo(Person other) {  
        return this.name.compareTo(other.name);  
    }  
}
```

- **Options:**

- A. It compares Person objects based on their names.
- B. It will cause a compilation error.
- C. It compares Person objects based on their hash codes.
- D. It is not a valid implementation of Comparable.

- **Correct Answer:** A. It compares Person objects based on their names.

*Explanation: The compareTo() method compares Person objects based on their name property.*

---

7. Which of the following statements is true about the natural order of a class that implements Comparable?

- **Options:**

- A. The natural order must be consistent with equals().
- B. The natural order must be defined in a separate Comparator class.
- C. The natural order cannot be overridden.
- D. The natural order is determined at runtime.

- **Correct Answer:** A. The natural order must be consistent with equals().

*Explanation: The natural order of a class that implements Comparable should be consistent with equals(), meaning if two objects are considered equal according to compareTo(), equals() should return true.*

---



**8. What happens if a class that implements Comparable does not provide a consistent compareTo method?**

- **Options:**
    - A. The class will fail to compile.
    - B. Sorting operations may produce incorrect results or throw exceptions.
    - C. The compareTo method will be ignored.
    - D. The equals method will override compareTo.
  - **Correct Answer:** B. Sorting operations may produce incorrect results or throw exceptions.  
*Explanation: If compareTo() is inconsistent (e.g., it doesn't properly order objects), sorting operations might yield incorrect results or even throw exceptions.*
- 

**9. What is the primary reason to implement Comparable in a class?**

- **Options:**
    - A. To allow objects of the class to be printed.
    - B. To define a natural ordering for objects of the class.
    - C. To enable deep cloning of objects of the class.
    - D. To enforce encapsulation of class fields.
  - **Correct Answer:** B. To define a natural ordering for objects of the class.  
*Explanation: The main purpose of implementing Comparable is to define the natural ordering of objects, allowing them to be compared and sorted.*
- 

**10. Given the following class definition, what would be the result of sorting a list of Product objects?**

java

Copy code

```
class Product implements Comparable<Product> {  
  
    String name;  
  
    double price;  
  
    Product(String name, double price) {  
        this.name = name;  
        this.price = price;  
    }  
}
```

```
@Override  
  
public int compareTo(Product other) {  
    return Double.compare(this.price, other.price);  
}  
}
```

- **Options:**

- A. Products will be sorted alphabetically by name.
- B. Products will be sorted by price in ascending order.
- C. Products will be sorted by price in descending order.
- D. Sorting will throw a ClassCastException.

- **Correct Answer:** B. Products will be sorted by price in ascending order.

*Explanation: The compareTo() method compares Product objects by their price, so sorting will order them by price in ascending order.*

1. **Which of the following classes in Java implements the Comparable interface to define natural ordering?**

- A. **HashMap**
- B. **ArrayList**
- C. **String**
- D. **LinkedList**

**Correct Answer: C. String**

*Explanation: The String class implements the Comparable interface to define its natural ordering based on lexicographical order.*

---

2. **What method must a class implement to define its natural order?**

- A. **compareTo()**
- B. **compare()**
- C. **sort()**
- D. **order()**

**Correct Answer: A. compareTo()**

*Explanation: To define the natural order of a class, it must implement the compareTo() method from the Comparable interface.*

---

3. **What will be the output of the following code?**

java

Copy code

```
List<Integer> list = Arrays.asList(3, 2, 1);
```

```
Collections.sort(list);
```

```
System.out.println(list);
```

- ☐ A. [1, 2, 3]
- ☐ B. [3, 2, 1]
- ☐ C. [2, 3, 1]
- ☐ D. [1, 3, 2]

**Correct Answer: A. [1, 2, 3]**

*Explanation:* The Collections.sort() method sorts the list in ascending order, so the output will be [1, 2, 3].

---

4. How does the compareTo method in the Comparable interface signal that one object is greater than, equal to, or less than another?

- ☐ A. By returning a boolean value
- ☐ B. By throwing an exception
- ☐ C. By returning an integer
- ☐ D. By calling another method

**Correct Answer: C. By returning an integer**

*Explanation:* The compareTo() method returns an integer: a negative value if the object is less than the other, zero if they are equal, and a positive value if the object is greater.

---

5. Which class in Java uses the natural ordering of its elements?

- ☐ A. HashSet
- ☐ B. TreeSet
- ☐ C. LinkedList
- ☐ D. Vector

**Correct Answer: B. TreeSet**

*Explanation:* The TreeSet class uses the natural ordering of its elements, or an ordering defined by a Comparator.

---

6. Given the following class definition, what is true about the compareTo method?

java

Copy code

```
public class Person implements Comparable<Person> {  
    private String name;  
  
    @Override  
    public int compareTo(Person other) {  
        return this.name.compareTo(other.name);  
    }  
}
```

- ☐ A. It compares Person objects based on their names.
- ☐ B. It will cause a compilation error.
- ☐ C. It compares Person objects based on their hash codes.
- ☐ D. It is not a valid implementation of Comparable.

**Correct Answer: A. It compares Person objects based on their names.**

*Explanation:* The compareTo() method compares Person objects based on their name property.

---

7. Which of the following statements is true about the natural order of a class that implements Comparable?

- ☐ A. The natural order must be consistent with equals().
- ☐ B. The natural order must be defined in a separate Comparator class.
- ☐ C. The natural order cannot be overridden.
- ☐ D. The natural order is determined at runtime.

**Correct Answer: A. The natural order must be consistent with equals().**

*Explanation:* The natural order of a class that implements Comparable should be consistent with equals(), meaning if two objects are considered equal according to compareTo(), equals() should return true.

---

8. What happens if a class that implements Comparable does not provide a consistent compareTo method?

- ☐ A. The class will fail to compile.
- ☐ B. Sorting operations may produce incorrect results or throw exceptions.
- ☐ C. The compareTo method will be ignored.

- D. The equals method will override compareTo.

**Correct Answer: B. Sorting operations may produce incorrect results or throw exceptions.**

*Explanation:* If compareTo() is inconsistent (e.g., it doesn't properly order objects), sorting operations might yield incorrect results or even throw exceptions.

---

**9. What is the primary reason to implement Comparable in a class?**

- A. To allow objects of the class to be printed.
- B. To define a natural ordering for objects of the class.
- C. To enable deep cloning of objects of the class.
- D. To enforce encapsulation of class fields.

**Correct Answer: B. To define a natural ordering for objects of the class.**

*Explanation:* The main purpose of implementing Comparable is to define the natural ordering of objects, allowing them to be compared and sorted.

---

**10. Given the following class definition, what would be the result of sorting a list of Product objects?**

java

Copy code

```
class Product implements Comparable<Product> {  
    String name;  
    double price;  
  
    Product(String name, double price) {  
        this.name = name;  
        this.price = price;  
    }  
  
    @Override  
    public int compareTo(Product other) {  
        return Double.compare(this.price, other.price);  
    }  
}
```

- A. Products will be sorted alphabetically by name.

- B. Products will be sorted by price in ascending order.
- C. Products will be sorted by price in descending order.
- D. Sorting will throw a `ClassCastException`.

**Correct Answer: B. Products will be sorted by price in ascending order.**

*Explanation:* The `compareTo()` method compares `Product` objects by their price, so sorting will order them by price in ascending order.

1. What is a weighted substring in the context of string processing?

- A) A substring that appears multiple times in the string.
- B) A substring that has a weight assigned based on a specific criterion.
- C) A substring that is longer than a certain length.
- D) A substring that contains only vowels.

**Correct Answer: B) A substring that has a weight assigned based on a specific criterion.**

---

2. Given a string "abc" with weights {a: 1, b: 2, c: 3}, what is the weight of the substring "ab"?

- A) 1
- B) 2
- C) 3
- D) 4

**Correct Answer: D) 4**

*Explanation:* The weight of the substring "ab" is the sum of the weights of 'a' (1) and 'b' (2), giving a total of 3.

---

3. In the string "abcd", if each character has a weight equal to its position (1-based index), what is the weight of the substring "cd"?

- A) 3
- B) 4
- C) 7
- D) 5

**Correct Answer: C) 7**

*Explanation:* The weight of the substring "cd" is the sum of the weights of 'c' (3) and 'd' (4), giving a total of 7.

---

4. Which algorithm can be used to efficiently find the maximum weighted substring in a string?

- A) Brute force

- **B) Dynamic programming**
- **C) Greedy algorithm**
- **D) Depth-first search**

**Correct Answer: B) Dynamic programming**

*Explanation:* Dynamic programming is efficient in solving problems related to finding the maximum weighted substring by breaking the problem into overlapping subproblems.

---

5. If you have a string "xyz" and weights {x: 5, y: 4, z: 3}, what is the weight of the substring "xyz"?

- **A) 12**
- **B) 9**
- **C) 15**
- **D) 10**

**Correct Answer: A) 12**

*Explanation:* The weight of the substring "xyz" is the sum of the weights of 'x' (5), 'y' (4), and 'z' (3), giving a total of 12.

---

6. Given a string "aab" with weights {a: 2, b: 3}, what is the maximum weighted substring?

- **A) "a"**
- **B) "aa"**
- **C) "b"**
- **D) "aab"**

**Correct Answer: B) "aa"**

*Explanation:* The maximum weighted substring is "aa", with a total weight of 4 (2 + 2).

---

7. In the context of weighted substrings, what does the term "prefix sum" refer to?

- **A) The sum of weights of all characters up to a given index.**
- **B) The sum of weights of all characters in the string.**
- **C) The sum of weights of all suffixes.**
- **D) The sum of weights of all unique substrings.**

**Correct Answer: A) The sum of weights of all characters up to a given index.**

*Explanation:* The prefix sum refers to the sum of weights of all characters up to a specific index in the string.

---

8. What is the time complexity of calculating the weight of all substrings using a brute force approach?

- A)  $O(n)$
- B)  $O(n^2)$
- C)  $O(n^3)$
- D)  $O(n^4)$

**Correct Answer: B)  $O(n^2)$**

*Explanation:* To find the weight of all substrings, we generate all pairs of start and end indices of substrings, which results in  $O(n^2)$  complexity.

---

9. For the string "pqr" with weights {p: 7, q: 6, r: 5}, what is the total weight of all possible substrings?

- A) 33
- B) 42
- C) 36
- D) 44

**Correct Answer: B) 42**

*Explanation:* The total weight is the sum of weights for all substrings: "p" (7), "q" (6), "r" (5), "pq" (13), "qr" (11), "pqr" (18). The total is  $7 + 6 + 5 + 13 + 11 + 18 = 42$ .

---

10. Which data structure can help in efficiently finding the maximum weighted substring in a large string?

- A) Array
- B) Linked List
- C) Segment Tree
- D) Queue

**Correct Answer: C) Segment Tree**

*Explanation:* A segment tree is an efficient data structure that helps in answering range queries and can be used to efficiently find the maximum weighted substring in large strings.

1. Which pivot selection strategy can help in avoiding the worst-case scenario in Quick Sort?

- A) Always pick the first element as pivot
- B) Always pick the last element as pivot
- C) Pick a random element as pivot
- D) Always pick the middle element as pivot

**Correct Answer: C) Pick a random element as pivot**



*Explanation:* Picking a random element as pivot helps avoid the worst-case scenario ( $O(n^2)$ ) in case of already sorted or reverse-sorted arrays.

---

2. In Quick Sort, the process of dividing the array into two parts is called:

- A) Merging
- B) Dividing
- C) Partitioning
- D) Conquering

**Correct Answer: C) Partitioning**

*Explanation:* Partitioning is the process of dividing the array into two parts based on the pivot element, where elements smaller than the pivot go to one side and elements greater go to the other.

---

3. Which of the following methods is used in Quick Sort to choose a pivot element?

- A) Median of three
- B) First element
- C) Random element
- D) All of the above

**Correct Answer: D) All of the above**

*Explanation:* Quick Sort can use various methods like median of three, first element, or a random element for selecting the pivot.

---

4. What is the primary advantage of using Quick Sort over other sorting algorithms?

- A) Simplicity of implementation
- B) Stability
- C) In-place sorting
- D) Requires less memory

**Correct Answer: C) In-place sorting**

*Explanation:* Quick Sort is an in-place sorting algorithm, meaning it doesn't require extra space for storing elements.

---

5. In Quick Sort, if the input array is already sorted in ascending order, which pivot selection strategy is most likely to cause the worst-case scenario?

- A) First element
- B) Last element

- C) Middle element
- D) Random element

**Correct Answer: A) First element**

*Explanation:* If the first element is selected as the pivot in a sorted array, it leads to the worst-case scenario ( $O(n^2)$ ) because the array is already partitioned in a skewed way.

---

6. Which of the following statements is true about the partitioning process in Quick Sort?

- A) The pivot element is always placed at the start of the array.
- B) The pivot element is always placed at the end of the array.
- C) The pivot element is always placed in its correct sorted position.
- D) The pivot element is placed in the middle of the array.

**Correct Answer: C) The pivot element is always placed in its correct sorted position.**

*Explanation:* After the partitioning process, the pivot element is placed in its correct position, and the elements on the left are smaller, while those on the right are larger.

---

7. What is a common strategy to handle the case when the input array has many duplicate elements in Quick Sort?

- A) Use a stable version of Quick Sort
- B) Use three-way partitioning
- C) Use the first element as pivot
- D) Switch to Selection Sort

**Correct Answer: B) Use three-way partitioning**

*Explanation:* Three-way partitioning is used to handle duplicates efficiently by splitting the array into three parts: less than, equal to, and greater than the pivot.

---

8. What is a key difference between Quick Sort and Merge Sort?

- A) Quick Sort is stable, Merge Sort is not
- B) Quick Sort is in-place, Merge Sort is not
- C) Quick Sort always uses the first element as pivot, Merge Sort uses the middle element
- D) Quick Sort divides the array into two equal halves, Merge Sort does not

**Correct Answer: B) Quick Sort is in-place, Merge Sort is not**

*Explanation:* Quick Sort is an in-place algorithm, while Merge Sort requires additional space to store the subarrays.

---

9. In which of the following scenarios is Quick Sort preferred over Merge Sort?

- A) When a stable sort is required
- B) When the input array is large and partially sorted
- C) When in-place sorting is required
- D) When the input array is small and random

**Correct Answer: C) When in-place sorting is required**

*Explanation:* Quick Sort is preferred when in-place sorting is required because it doesn't need extra memory like Merge Sort.

---

10. Which of the following is a correct way to implement the partition function in Quick Sort?

- A) Hoare partition scheme
- B) Lomuto partition scheme
- C) Median-of-three partition scheme
- D) All of the above

**Correct Answer: D) All of the above**

*Explanation:* There are different partition schemes in Quick Sort, including Hoare and Lomuto, and the median-of-three method for pivot selection.

1. Which of the following is the correct way to move all hyphens to the beginning of a string in Java?

- A) String result = str.replace("-", "") + str.replaceAll("[^]", "");
- B) String result = str.replaceAll("-", "") + str.replaceAll("[^]", "");
- C) String result = str.replaceAll("[^]", "") + str.replaceAll("-", "");
- D) String result = str.replaceAll("-", "") + str.replaceAll("[^]", "");

**Correct Answer: D) String result = str.replaceAll("-", "") + str.replaceAll("[^]", "");**

*Explanation:* The correct approach is to remove hyphens first and then append them after all non-hyphen characters are removed.

---

2. Given a string str = "a-b-c-d", what will be the result of the following code?

java

Copy code

```
String result = str.replaceAll("-", "") + str.replaceAll("[^]", "");
```

- A) abcd----
- B) ----abcd
- C) a--b--cd

- **D) abcd**

**Correct Answer: A) abcd----**

*Explanation:* First, `str.replaceAll("-", "")` removes all hyphens, producing "abcd", and `str.replaceAll("[^~]", "")` removes all non-hyphen characters, leaving "----". The result is "abcd----".

---

3. **What is the time complexity of moving hyphens to the beginning of a string using the `replaceAll` method in Java?**

- **A)  $O(n)$**
- **B)  $O(n^2)$**
- **C)  $O(n \log n)$**
- **D)  $O(1)$**

**Correct Answer: B)  $O(n^2)$**

*Explanation:* The `replaceAll` method involves regular expression evaluation and string copying, which results in a time complexity of  $O(n^2)$  due to the intermediate string creation.

---

4. **Which of the following methods can be used to move all hyphens to the beginning of a string without using `replaceAll`?**

- **A) `StringBuilder`**
- **B) `StringBuffer`**
- **C) `String`**
- **D) `Array`**

**Correct Answer: A) `StringBuilder`**

*Explanation:* `StringBuilder` can be used to efficiently build the string by appending and prepending characters in a loop.

---

5. **Given the string `str = "a-b-c-d"`, which of the following snippets correctly moves all hyphens to the beginning?**

java

Copy code

```
StringBuilder sb = new StringBuilder();
for (char ch : str.toCharArray()) {
    if (ch == '-') {
        sb.insert(0, ch);
    } else {
```

```
        sb.append(ch);
    }
}
```

- ☐ A) sb.toString() results in ----abcd
- ☐ B) sb.toString() results in abcd----
- ☐ C) sb.toString() results in a--b--cd
- ☐ D) sb.toString() results in abcd

**Correct Answer: A)** sb.toString() results in ----abcd

*Explanation:* This approach adds hyphens at the start of the StringBuilder and appends the non-hyphen characters at the end, resulting in all hyphens moved to the beginning.

---

6. Which of the following regex patterns correctly matches all non-hyphen characters in a string?

- ☐ A) [^~]
- ☐ B) [-]
- ☐ C) [^a-zA-Z0-9]
- ☐ D) [a-zA-Z0-9]

**Correct Answer: A)** [^~]

*Explanation:* The regex [^~] matches all characters that are not hyphens.

---

7. Given a string str = "-a-b-c-d-", what will be the result of the following code?

java

Copy code

```
String result = str.replaceAll("-", "") + str.replaceAll("[^~]", "");
```

- ☐ A) abcd----
- ☐ B) ----abcd
- ☐ C) a--b--c--d
- ☐ D) ----abcd----

**Correct Answer: B)** ----abcd

*Explanation:* First, str.replaceAll("-", "") removes all hyphens, producing "abcd", and str.replaceAll("[^~]", "") removes all non-hyphen characters, leaving "----". The result is "----abcd".

---

8. Which of the following Java collections is most suitable for storing the hyphens while iterating through the string?

- A) ArrayList
- B) LinkedList
- C) HashSet
- D) TreeSet

**Correct Answer: B) LinkedList**

*Explanation:* LinkedList is most suitable for operations like inserting elements at the beginning due to its efficient insertions.

---

9. What will be the output of the following code snippet?

java

Copy code

```
String str = "a-b-c-d";
int count = 0;
StringBuilder sb = new StringBuilder();
for (char ch : str.toCharArray()) {
    if (ch == '-') {
        count++;
    } else {
        sb.append(ch);
    }
}
for (int i = 0; i < count; i++) {
    sb.insert(0, '-');
}
System.out.println(sb.toString());
```

- A) abcd----
- B) ----abcd
- C) a--b--cd
- D) abcd

**Correct Answer: B) ----abcd**

*Explanation:* This code counts the hyphens, removes them, and then inserts the same number of hyphens at the beginning.

---

10. What does the `replaceAll("[^-]", "")` part of the code do?

- A) Replaces all hyphens with empty strings
  - B) Removes all hyphens from the string
  - C) Removes all non-hyphen characters from the string
  - D) Replaces all non-hyphen characters with empty strings
- Correct Answer: D)** Replaces all non-hyphen characters with empty strings

*Explanation:* The `replaceAll("[^-]", "")` removes all characters that are not hyphens.

1. What problem does Manacher's algorithm solve efficiently?

- A) Longest common subsequence
- B) Longest palindromic substring
- C) Shortest path in a graph
- D) Sorting an array

**Answer: B) Longest palindromic substring**

---

2. Which data structure is primarily used in Manacher's algorithm?

- A) Stack
- B) Queue
- C) Hash table
- D) None of the above

**Answer: D) None of the above**

---

3. What is the time complexity of Manacher's algorithm?

- A)  $O(n)$
- B)  $O(n \log n)$
- C)  $O(n^2)$
- D)  $O(n^3)$

**Answer: A)  $O(n)$**

---

4. Which of the following is a preprocessing step in Manacher's algorithm?

- A) Sorting the characters

- B) Calculating the prefix function
- C) Reversing the string
- D) None of the above

**Answer: D) None of the above**

---

5. In Manacher's algorithm, what does the array P[] represent?

- A) Prefix function
- B) Longest palindromic substring lengths
- C) Suffix array
- D) None of the above

**Answer: B) Longest palindromic substring lengths**

---

6. Which technique is used to handle even-length palindromes in Manacher's algorithm?

- A) Dynamic programming
- B) Two-pointer technique
- C) Extension by one character
- D) None of the above

**Answer: C) Extension by one character**

---

7. What is the space complexity of Manacher's algorithm?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(n^2)$
- D)  $O(n \log n)$

**Answer: B)  $O(n)$**

---

8. Which of the following is NOT a characteristic of Manacher's algorithm?

- A) Efficiently finds all palindromic substrings
- B) Requires  $O(n)$  time complexity
- C) Utilizes dynamic programming
- D) Does not require any additional data structures

**Answer: C) Utilizes dynamic programming**

---



9. What is the main advantage of Manacher's algorithm over a naive approach to find palindromic substrings?

- A) It guarantees finding the longest palindromic substring
- B) It has a lower time complexity
- C) It uses less memory
- D) It can handle only odd-length palindromes efficiently

**Answer: B) It has a lower time complexity**

---

10. In which year was Manacher's algorithm proposed?

- A) 1975
- B) 1989
- C) 1995
- D) 2005

**Answer: B) 1989**

1. What is a sorted unique permutation of a string?

- A) A permutation where characters are sorted in descending order
- B) A permutation where characters are sorted in ascending order and no characters are repeated
- C) A permutation where characters are in any order and can repeat
- D) None of the above

**Answer: B) A permutation where characters are sorted in ascending order and no characters are repeated**

---

2. How many sorted unique permutations can be generated from the string "ABC"?

- A) 3
- B) 6
- C) 9
- D) 12

**Answer: B) 6**

---

3. Which algorithmic technique is commonly used to generate sorted unique permutations efficiently?

- A) Depth-first search (DFS)
- B) Breadth-first search (BFS)

- C) Greedy algorithm
- D) Dynamic programming

**Answer: A) Depth-first search (DFS)**

---

4. **What is the time complexity for generating all sorted unique permutations of a string of length  $n$  using backtracking?**

- A)  $O(n)$
- B)  $O(n \log n)$
- C)  $O(n!)$
- D)  $O(2^n)$

**Answer: C)  $O(n!)$**

---

5. **In which order are sorted unique permutations typically generated using backtracking?**

- A) Random order
- B) Reverse lexicographical order
- C) Lexicographical order
- D) No specific order

**Answer: C) Lexicographical order**

---

6. **Which data structure is suitable for efficiently checking and maintaining used characters during permutation generation?**

- A) Stack
- B) Queue
- C) Set
- D) Array

**Answer: C) Set**

---

7. **What is the key property of sorted unique permutations that distinguishes them from all possible permutations?**

- A) They are sorted in descending order
- B) They are sorted in ascending order
- C) They contain all characters of the original string
- D) They have the maximum length possible

**Answer: B) They are sorted in ascending order**

---

8. Which approach is typically used to avoid generating duplicate permutations in algorithms that generate all permutations?

- A) Hashing
- B) Sorting
- C) Backtracking
- D) Dynamic programming

**Answer: C) Backtracking**

---

9. Which of the following statements about sorted unique permutations is true?

- A) They are always of the same length as the original string
- B) They are always longer than the original string
- C) They are always shorter than the original string
- D) None of the above

**Answer: A) They are always of the same length as the original string**

---

10. What is the space complexity for generating all sorted unique permutations of a string of length  $n$  using backtracking?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(n!)$
- D)  $O(2^n)$

**Answer: C)  $O(n!)$**

---

1. In military terms, what does the term "maneuvering" primarily refer to?

- A) Setting up camp
- B) Planning logistics
- C) Movement and positioning of forces
- D) Negotiating peace treaties

**Answer: C) Movement and positioning of forces**

---

2. Which of the following is a key principle of defensive maneuvering in driving?

- A) Tailgating
- B) Anticipating hazards

- C) Speeding up to pass obstacles
- D) Ignoring traffic signals

**Answer: B) Anticipating hazards**

---

**3. What is a common strategy in aerial combat maneuvering (dogfighting)?**

- A) Flying straight at the enemy
- B) Staying at a constant altitude
- C) Maintaining a predictable flight path
- D) Using evasive maneuvers

**Answer: D) Using evasive maneuvers**

---

**4. Which term is used to describe a sudden change in direction or tactic to gain an advantage in a competitive situation?**

- A) Pivot
- B) Retreat
- C) Maneuver
- D) Stagnation

**Answer: C) Maneuver**

---

**5. What is the primary goal of offensive maneuvering in military strategy?**

- A) Protecting supply lines
- B) Holding defensive positions
- C) Gaining a tactical advantage
- D) Avoiding engagements

**Answer: C) Gaining a tactical advantage**

---

**6. Which driving maneuver involves reversing the vehicle to change its direction?**

- A) U-turn
- B) Parallel parking
- C) Three-point turn
- D) J-turn

**Answer: C) Three-point turn**

---

7. In naval warfare, what does "maneuvering" typically refer to?

- A) Firing cannons
- B) Sailing tactics
- C) Anchoring ships
- D) Repairing hulls

**Answer: B) Sailing tactics**

---

8. Which factor is crucial for effective maneuvering in team sports such as basketball or soccer?

- A) Individual scoring ability
- B) Team communication
- C) Timekeeping
- D) Spectator support

**Answer: B) Team communication**

---

9. What is a primary consideration for safe maneuvering in mountain climbing?

- A) Climbing in a straight line
- B) Speeding up to reach the summit faster
- C) Using proper safety equipment
- D) Climbing alone

**Answer: C) Using proper safety equipment**

---

10. Which of the following statements about maneuvering is true across various contexts?

- A) It always involves aggressive actions
- B) It can only be strategic, not tactical
- C) It requires adaptability and quick decision-making
- D) It is a slow and deliberate process

**Answer: C) It requires adaptability and quick decision-making**

1. In how many ways can you choose 3 books from a shelf containing 7 different books?

- A) 21
- B) 35
- C) 42

- D) 84

**Answer: A) 21**

(This is a combination problem:  $\binom{7}{3} = \frac{7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{3 \times 2 \times 1} = 21$ )

---

2. A committee of 5 people is to be formed from a group of 10 individuals. How many different committees can be formed?

- A) 252
- B) 210
- C) 120
- D) 2520

**Answer: B) 210**

(This is a combination problem:  $\binom{10}{5} = \frac{10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{5 \times 4 \times 3 \times 2 \times 1} = 210$ )

---

3. How many different 4-letter combinations can be formed from the letters A, B, C, D, E, F without repetition?

- A) 120
- B) 360
- C) 720
- D) 24

**Answer: B) 360**

(This is a permutation problem:  $P(6,4) = \frac{6!}{(6-4)!} = \frac{6 \times 5 \times 4 \times 3 \times 2 \times 1}{2 \times 1} = 360$ )

---

4. How many ways can you arrange the letters in the word "COMBINATIONS"?

- A) 5040
- B) 3628800
- C) 40320
- D) 1440

**Answer: B) 3628800**

(There are 11 letters with repeated "O", "I", "N", "T" which result in this calculation:  $\frac{11!}{2!2!2!2!} = \frac{39810100}{16} = 2488131.25$ )

---

5. In a lottery, 5 numbers are to be picked from 1 to 50. How many different combinations of numbers can be chosen?

- A) 2118760
- B) 1024
- C) 3125
- D) 2598960

**Answer: A) 2118760**

(This is a combination problem:

$(505) = 50 \times 49 \times 48 \times 47 \times 46 \times 45 \times 44 \times 43 \times 42 \times 41 = 2118760$   
 $\binom{50}{5} = \frac{50 \times 49 \times 48 \times 47 \times 46}{5 \times 4 \times 3 \times 2 \times 1} = 2118760$   
 $(550) = 55 \times 54 \times 53 \times 52 \times 51 \times 50 \times 49 \times 48 \times 47 \times 46 \times 45 \times 44 \times 43 \times 42 \times 41 \times 40 \times 39 \times 38 \times 37 \times 36 \times 35 \times 34 \times 33 \times 32 \times 31 \times 30 \times 29 \times 28 \times 27 \times 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 20 \times 19 \times 18 \times 17 \times 16 \times 15 \times 14 \times 13 \times 12 \times 11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 550!$   
 $\binom{55}{5} = \frac{55 \times 54 \times 53 \times 52 \times 51}{5 \times 4 \times 3 \times 2 \times 1} = 2598960$

---

6. How many ways can you select 2 red marbles and 3 blue marbles from a collection of 10 red and 15 blue marbles?

- A) 210
- B) 3003
- C) 252
- D) 126

**Answer: C) 252**

(This is a combination problem:  $\binom{10}{2} \times \binom{15}{3} = 45 \times 455 = 252$   
 $\binom{10}{2} = \frac{10 \times 9}{2 \times 1} = 45$   
 $\binom{15}{3} = \frac{15 \times 14 \times 13}{3 \times 2 \times 1} = 455$   
 $45 \times 455 = 252$ )

---

7. A password consists of 4 digits where the first digit cannot be 0. How many different passwords are possible?

- A) 9000
- B) 7290
- C) 6480
- D) 6561

**Answer: A) 9000**

(For the first digit, there are 9 options (1-9) and for each of the next three digits, there are 10 options (0-9):  $9 \times 10 \times 10 \times 10 = 9000$   
 $9 \times 10 \times 10 \times 10 = 9000$ )

---

8. How many different 7-digit telephone numbers can be formed if each number must start with 5 and no digit can be repeated?

- A) 720
- B) 5040
- C) 362880

- D) 1440

**Answer: A) 720**

(The first digit is fixed as 5, so there are 9 options for the second digit, 8 options for the third, and so on:  $9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 = 720$   $\times 8 \times 7 \times 6 \times 5 \times 4 \times 3 = 720$ )

---

9. In how many ways can you distribute 8 identical chocolates among 4 children such that each child gets at least one chocolate?

- A) 70
- B) 35
- C) 56
- D) 84

**Answer: A) 70**

(This is a stars and bars problem:  $(8-4+1) = 5$   $\times \binom{5-1}{4-1} = \binom{4}{3} = 4$   $\times \binom{4-1}{3-1} = \binom{3}{2} = 3$   $\times \binom{3-1}{2-1} = \binom{2}{1} = 2$   $\times \binom{2-1}{1-1} = \binom{1}{0} = 1$   $\times 70 = 70$ )

---

10. How many different ways can you arrange the letters in the word "SUCCESS"?

- A) 2520
- B) 420
- C) 720
- D) 210

**Answer: B) 420**

(There are 7 letters with repeated "S" (3 times) and "C" (2 times):  $\frac{7!}{3!2!} = \frac{5040}{6 \times 2} = 420$ )

1. In the classic Josephus problem where every second person is eliminated in a circle until only one remains, if there are 12 people, which position would be the last person standing?

- A) 7
- B) 9
- C) 11
- D) 12

**Answer: B) 9**

(Using the Josephus problem formula, the position of the last person standing for 12 people where every second person is eliminated is 9.)

---



2. In a circle of 20 people playing the Josephus game (every 3rd person is eliminated), what position would the last remaining person be?

- A) 9
- B) 12
- C) 16
- D) 19

**Answer: C) 16**

(The Josephus problem for 20 people and eliminating every 3rd person results in position 16 being the last remaining person.)

---

3. The Josephus problem is a classic theoretical problem in:

- A) Graph theory
- B) Number theory
- C) Game theory
- D) Combinatorics

**Answer: C) Game theory**

(The Josephus problem is a problem in game theory that involves elimination and strategy in a circular setup.)

---

4. If there are 15 people standing in a circle and every 4th person is eliminated until only one remains, which position is safe from elimination?

- A) 9
- B) 11
- C) 12
- D) 15

**Answer: B) 11**

(For 15 people, eliminating every 4th person, position 11 is the safe one.)

---

5. In the Josephus problem with 10 people where every 2nd person is eliminated, what is the safe position?

- A) 4
- B) 5
- C) 7
- D) 10

**Answer: B) 5**

(In the Josephus problem for 10 people with every second person eliminated, position 5 is the safe one.)

---

6. **The Josephus problem is closely related to which branch of mathematics?**

- A) Algebra
- B) Geometry
- C) Number theory
- D) Probability theory

**Answer: C) Number theory**

(The Josephus problem involves understanding patterns and relationships, which are closely tied to number theory.)

---

7. **If there are 25 people in a circle and every 5th person is eliminated, which position will be the last remaining?**

- A) 13
- B) 17
- C) 19
- D) 23

**Answer: B) 17**

(For 25 people, eliminating every 5th person, position 17 is the last remaining person.)

---

8. **In the Josephus problem with 8 people and every 3rd person being eliminated, which position is the last remaining person?**

- A) 4
- B) 6
- C) 7
- D) 8

**Answer: C) 7**

(With 8 people and eliminating every 3rd person, position 7 is the last remaining.)

---

9. **The Josephus problem finds applications in:**

- A) Cryptography
- B) Sorting algorithms
- C) Data compression

- D) None of the above

**Answer: D) None of the above**

(The Josephus problem is a theoretical problem used to demonstrate elimination patterns and does not have direct applications in cryptography, sorting, or data compression.)

---

10. If there are 30 people standing in a circle and every 6th person is eliminated, which position will be the last person standing?

- A) 12
- B) 16
- C) 18
- D) 30

**Answer: B) 16**

(For 30 people and eliminating every 6th person, position 16 will be the last remaining person.)

1. What is a common algorithm used for solving mazes with a single solution path?

- A) Depth-First Search (DFS)
- B) Breadth-First Search (BFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: A) Depth-First Search (DFS)**

(DFS is commonly used for maze-solving when there is a single solution path, as it explores each path deeply before backtracking.)

---

2. Which data structure is typically used to implement DFS and BFS for maze solving?

- A) Queue
- B) Stack
- C) Priority Queue
- D) Linked List

**Answer: A) Queue** (for BFS)

**Answer: B) Stack** (for DFS)

(BFS uses a queue, while DFS uses a stack to manage the exploration of nodes in the maze.)

---

3. Which algorithm guarantees finding the shortest path in an unweighted maze?

- A) Depth-First Search (DFS)

- B) Breadth-First Search (BFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: B) Breadth-First Search (BFS)**

(BFS guarantees the shortest path in an unweighted maze as it explores all nodes level by level.)

---

4. Which heuristic is commonly used in the A search algorithm for maze solving?\*

- A) Manhattan distance
- B) Euclidean distance
- C) Hamming distance
- D) Chebyshev distance

**Answer: A) Manhattan distance**

(Manhattan distance is commonly used in grid-based mazes, where movements are restricted to horizontal and vertical directions.)

---

5. In maze solving, what does the term "backtracking" refer to?

- A) A method to find dead-ends
- B) Reversing the path taken
- C) A systematic way to explore paths
- D) Eliminating invalid paths

**Answer: B) Reversing the path taken**

(Backtracking refers to retracing steps to explore alternative paths after reaching a dead-end.)

---

6. Which traversal technique is more suitable for finding all possible solutions in a maze?

- A) Depth-First Search (DFS)
- B) Breadth-First Search (BFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: A) Depth-First Search (DFS)**

(DFS explores as deeply as possible into the maze, thus allowing it to find all possible solutions.)

---

7. Which algorithm is preferred for finding the shortest path in a weighted maze?

- A) Depth-First Search (DFS)
- B) Breadth-First Search (BFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: C) Dijkstra's Algorithm**

(Dijkstra's algorithm is specifically designed to handle weighted graphs, making it suitable for weighted mazes.)

---

**8. What is the time complexity of Breadth-First Search (BFS) for maze solving?**

- A)  $O(V)$
- B)  $O(E)$
- C)  $O(V + E)$
- D)  $O(V^2)$

**Answer: C)  $O(V + E)$**

(The time complexity of BFS is  $O(V + E)$ , where  $V$  is the number of vertices and  $E$  is the number of edges.)

---

**9. Which of the following algorithms is not suitable for maze solving that involves finding the shortest path?**

- A) Breadth-First Search (BFS)
- B) Depth-First Search (DFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: B) Depth-First Search (DFS)**

(DFS does not guarantee the shortest path, as it explores one path fully before backtracking, which may lead to suboptimal solutions.)

---

**10. Which algorithm is generally more efficient when a heuristic can be defined for the maze-solving problem?**

- A) Depth-First Search (DFS)
- B) Breadth-First Search (BFS)
- C) Dijkstra's Algorithm
- D) A\* Search Algorithm

**Answer: D) A Search Algorithm\***

(A\* search algorithm is designed to be efficient when a heuristic can be used to estimate the cost of reaching the goal.)

❓ In how many directions do queens attack each other?

- A) 1
- B) 2
- C) 3
- D) 4
- **Answer: D) 4**

❓ Placing n-queens so that no two queens attack each other is called?

- A) n-queen's problem
- B) 8-queen's problem
- C) Hamiltonian circuit problem
- D) Subset sum problem
- **Answer: A) n-queen's problem**

❓ Where is the n-queens problem implemented?

- A) Carom
- B) Chess
- C) Ludo
- D) Cards
- **Answer: B) Chess**

❓ In n-queen problem, how many values of n does not provide an optimal solution?

- A) 1
- B) 2
- C) 3
- D) 4
- **Answer: B) 2**  
(For  $n = 2$  and  $n = 3$ , there is no solution.)

❓ Which of the following methods can be used to solve n-queen's problem?

- A) Greedy algorithm
- B) Divide and conquer
- C) Iterative improvement
- D) Backtracking
- **Answer: D) Backtracking**

❑ Of the following given options, which one of the following is a correct option that provides an optimal solution for the 4-queens problem?

- A) (3,1,4,2)
- B) (2,3,1,4)
- C) (4,3,2,1)
- D) (4,2,3,1)
- **Answer: A) (3,1,4,2)**

❑ How many possible solutions exist for an 8-queen problem?

- A) 100
- B) 98
- C) 92
- D) 88
- **Answer: C) 92**

❑ How many possible solutions occur for a 10-queen problem?

- A) 850
- B) 742
- C) 842
- D) 724
- **Answer: D) 724**

❑ If  $n=1$ , an imaginary solution for the problem exists.

- A) True
- B) False
- **Answer: A) True**

❑ What is the domination number for 8-queen's problem?

- A) 8
- B) 7
- C) 6
- D) 5
- **Answer: A) 8**

❑ In the implementation of Warnsdorff's algorithm, what is the "tie-breaking" strategy typically used when multiple moves have the same degree?

- A) Choosing the move that leads to the highest degree in the subsequent move

- B) Choosing the move closest to the board's center
- C) Choosing the move that comes first in a predefined order
- D) Choosing the move with the highest potential degree
- **Answer: A) Choosing the move that leads to the highest degree in the subsequent move**

❓ **Why is Warnsdorff's algorithm considered a heuristic method rather than an exact algorithm?**

- A) Because it guarantees an optimal solution
- B) Because it may not always find a solution, especially for larger boards
- C) Because it requires backtracking to find a solution
- D) Because it is based on randomized decision-making
- **Answer: B) Because it may not always find a solution, especially for larger boards**

❓ **What is the role of the "degree of a vertex" in the context of Warnsdorff's algorithm?**

- A) It represents the number of possible moves a knight can make from that position
- B) It indicates the number of knights on the board
- C) It measures the difficulty of reaching the vertex
- D) It is the number of moves required to return to the starting position
- **Answer: A) It represents the number of possible moves a knight can make from that position**

❓ **In what way does Warnsdorff's algorithm utilize a greedy approach?**

- A) It optimizes the total path length of the Knight's Tour
- B) It prioritizes moves that minimize the risk of getting trapped early
- C) It ensures that the knight always moves to the closest available square
- D) It maximizes the number of moves made by the knight
- **Answer: B) It prioritizes moves that minimize the risk of getting trapped early**

❓ **What is the difference between an "open" and a "closed" Knight's Tour in the context of Warnsdorff's algorithm?**

- A) An open tour starts and ends on different squares, while a closed tour starts and ends on the same square
- B) An open tour uses all squares of the board, while a closed tour does not
- C) An open tour allows revisiting squares, while a closed tour does not
- D) An open tour is solved using Warnsdorff's algorithm, while a closed tour is not
- **Answer: A) An open tour starts and ends on different squares, while a closed tour starts and ends on the same square**



❑ **How can Warnsdorff's algorithm be modified to increase the likelihood of finding a solution on larger boards?**

- A) By implementing backtracking when a trap is detected
- B) By increasing the number of possible moves the knight can make
- C) By ignoring the degree heuristic in certain cases
- D) By ensuring the knight visits the center of the board frequently
- **Answer: A) By implementing backtracking when a trap is detected**

❑ **Which of the following is a known limitation of Warnsdorff's algorithm in practical implementations?**

- A) It requires exponential time to find a solution
- B) It can lead to incomplete tours due to local optimization
- C) It only works for 8x8 boards
- D) It fails to find a solution when starting from the corner of the board
- **Answer: B) It can lead to incomplete tours due to local optimization**

❑ **In computational complexity terms, why is Warnsdorff's algorithm often preferred over brute force approaches for the Knight's Tour problem?**

- A) It has a polynomial time complexity in comparison to the factorial time complexity of brute force methods
- B) It guarantees an optimal tour length
- C) It finds the shortest possible path for the knight
- D) It uses less memory than brute force approaches
- **Answer: A) It has a polynomial time complexity in comparison to the factorial time complexity of brute force methods**

❑ **When applied to non-standard board sizes (not 8x8), what is the effect on the efficiency and effectiveness of Warnsdorff's algorithm?**

- A) The algorithm becomes more efficient
- B) The algorithm's effectiveness can vary, sometimes failing to find a solution
- C) The algorithm's complexity decreases
- D) The algorithm always fails to find a solution
- **Answer: B) The algorithm's effectiveness can vary, sometimes failing to find a solution**

❑ **What is a common technique to improve the performance of Warnsdorff's algorithm?**

- A) Randomizing the initial knight's position
- B) Implementing a hybrid approach that combines it with backtracking

- C) Ignoring the least-constraining value heuristic
- D) Always starting the tour from the center of the board
- **Answer: B) Implementing a hybrid approach that combines it with backtracking**

🔍 **What distinguishes a Hamiltonian cycle from a Hamiltonian path?**

- A) A Hamiltonian cycle includes all vertices exactly once, while a Hamiltonian path may repeat vertices.
- B) A Hamiltonian cycle forms a closed loop, while a Hamiltonian path does not return to the starting vertex.
- C) A Hamiltonian cycle only exists in directed graphs, while a Hamiltonian path exists in undirected graphs.
- D) There is no difference; both terms refer to the same concept.
- **Answer: B) A Hamiltonian cycle forms a closed loop, while a Hamiltonian path does not return to the starting vertex.**

🔍 **In terms of computational complexity, the problem of finding a Hamiltonian cycle in a graph is classified as:**

- A) P
- B) NP
- C) NP-hard
- D) NP-complete
- **Answer: D) NP-complete**

🔍 **Which algorithmic approach is typically ineffective for finding Hamiltonian cycles in large graphs due to its exponential time complexity?**

- A) Dynamic programming
- B) Greedy algorithm
- C) Brute force
- D) Backtracking
- **Answer: C) Brute force**

🔍 **In the context of Hamiltonian cycles, what is the significance of a "complete graph"?**

- A) It guarantees the existence of a Hamiltonian cycle.
- B) It has a Hamiltonian path but not necessarily a Hamiltonian cycle.
- C) It has no Hamiltonian cycle if the number of vertices is odd.
- D) It guarantees that all vertices are connected by at least one edge.
- **Answer: A) It guarantees the existence of a Hamiltonian cycle.**

❑ Which of the following is NOT a necessary condition for the existence of a Hamiltonian cycle in an undirected graph?

- A) The graph must be connected.
- B) Every vertex must have at least two neighbors.
- C) The graph must be complete.
- D) The number of vertices with an odd degree must be even.
- **Answer: C) The graph must be complete.**

❑ Consider Dirac's theorem in graph theory. What does it state regarding the Hamiltonian cycle?

- A) A graph with  $n$  vertices is Hamiltonian if every vertex has degree at least  $n/2$ .
- B) A graph is Hamiltonian if it contains an Eulerian cycle.
- C) A graph is Hamiltonian if it contains a cycle of length  $n$ .
- D) A graph with  $n$  vertices is Hamiltonian if the sum of the degrees of any two adjacent vertices is at least  $n$ .
- **Answer: A) A graph with  $n$  vertices is Hamiltonian if every vertex has degree at least  $n/2$ .**

❑ In directed graphs, what special type of Hamiltonian cycle is called a "Hamiltonian circuit"?

- A) A cycle that includes a specific set of edges
- B) A cycle that covers all vertices and edges
- C) A cycle that has at least one repeated vertex
- D) A directed cycle that includes every vertex exactly once
- **Answer: D) A directed cycle that includes every vertex exactly once**

❑ Which of the following graph classes is known to always contain a Hamiltonian cycle?

- A) Planar graphs
- B) Bipartite graphs
- C) Complete graphs
- D) Trees
- **Answer: C) Complete graphs**

❑ What is the Travelling Salesman Problem (TSP) in relation to the Hamiltonian cycle?

- A) It is a problem of finding a path that covers all vertices without repeating.
- B) It is a variation where the Hamiltonian cycle needs to have the minimum possible weight.
- C) It is a problem of finding a cycle that visits each vertex exactly twice.
- D) It requires finding the shortest path between two specific vertices.

- **Answer: B) It is a variation where the Hamiltonian cycle needs to have the minimum possible weight.**

❑ **What is the Bondy–Chvátal theorem's contribution to the study of Hamiltonian cycles?**

- A) It provides a polynomial-time algorithm to find Hamiltonian cycles.
- B) It states that a graph is Hamiltonian if the sum of the degrees of any two non-adjacent vertices is at least the number of vertices in the graph.
- C) It provides a necessary and sufficient condition for the existence of Hamiltonian cycles in planar graphs.
- D) It defines a method for transforming non-Hamiltonian graphs into Hamiltonian graphs.
- **Answer: B) It states that a graph is Hamiltonian if the sum of the degrees of any two non-adjacent vertices is at least the number of vertices in the graph.**

❑ **Kruskal's algorithm is used to \_\_\_\_\_**

- A) find minimum spanning tree
- B) find single source shortest path
- C) find all pair shortest path algorithm
- D) traverse the graph
- **Answer: A) find minimum spanning tree**

❑ **Kruskal's algorithm is a \_\_\_\_\_**

- A) divide and conquer algorithm
- B) dynamic programming algorithm
- C) greedy algorithm
- D) approximation algorithm
- **Answer: C) greedy algorithm**

❑ **What is the weight of the minimum spanning tree using the Kruskal's algorithm?**

*(Note: Without the graph, I can't determine the exact weight. Please provide the graph if needed.)*

❑ **What is the time complexity of Kruskal's algorithm?**

- A)  $O(\log V)$
- B)  $O(E \log V)$
- C)  $O(E^2)$
- D)  $O(V \log E)$
- **Answer: B)  $O(E \log V)$**

❑ **Using Kruskal's algorithm, which edge will be selected first?**

*(Note: Without the graph, I can't determine the correct edge. Please provide the graph if needed.)*

❑ Which of the following edges form the minimum spanning tree on the graph using Kruskal's algorithm?

(Note: Without the graph, I can't determine the correct edges. Please provide the graph if needed.)

❑ Which of the following is true?

- A) Prim's algorithm can also be used for disconnected graphs
- B) Kruskal's algorithm can also run on the disconnected graphs
- C) Prim's algorithm is simpler than Kruskal's algorithm
- D) In Kruskal's sort edges are added to MST in decreasing order of their weights
- **Answer: B) Kruskal's algorithm can also run on the disconnected graphs**

❑ Which of the following is false about Kruskal's algorithm?

- A) It is a greedy algorithm
- B) It constructs MST by selecting edges in increasing order of their weights
- C) It can accept cycles in the MST
- D) It uses union-find data structure
- **Answer: C) It can accept cycles in the MST**

❑ Kruskal's algorithm is best suited for dense graphs than Prim's algorithm.

- A) True
- B) False
- **Answer: B) False**

❑ Consider the following statements.

- S1: Kruskal's algorithm might produce a non-minimal spanning tree.
- S2: Kruskal's algorithm can be efficiently implemented using the disjoint-set data structure.
- A) S1 is true but S2 is false
- B) Both S1 and S2 are false
- C) Both S1 and S2 are true
- D) S2 is true but S1 is false
- **Answer: D) S2 is true but S1 is false**

1. Which of the following approaches should be used to find the solution of the activity selection problem?

- A) Greedy approach
- B) Divide and conquer
- C) Brute-force approach

- D) Dynamic programming
  - **Answer: A) Greedy approach**
2. **Suppose two activities A and B, having start and finish time as  $SA$ ,  $FA$ , and  $SB$ ,  $FB$  respectively. Both the activities are said to be compatible, under which of the following conditions?**
- A)  $SA = FB$
  - B)  $SA > FB$
  - C)  $SA \geq FB$  or  $SB \geq FA$
  - D)  $SA \geq FB$  and  $SB = FA$
  - **Answer: D)  $SA \geq FB$  and  $SB = FA$**
3. **Consider the following algorithm to find the solution of the activity selection problem. Which of the following options is best suited to fill the blank?**
- Sort the given activity list  
display the first activity  
set  $i = 1$   
for  $j = 1$  to  $n-1$  do  
if begin time of activity[ $j$ ]  $\geq$  completion of activity[ $i$ ] then \_\_\_\_\_
    - A) display the  $i$ th activity
    - B) display the  $j$ th activity
    - C) discard the activity
    - D) increment  $j$
  - **Answer: B) display the  $j$ th activity**
4. **Consider the following number of activities with their start and finish time given below. In which sequence will the activity be selected in order to maximize the number of activities, without any conflicts?**  
*(Note: Without the graph, I can't determine the exact sequence. Please provide the graph if needed.)*
5. **Time complexity to find the solution of the activity selection problem takes  $O(n \log n)$  time, when the list is not sorted.**
- A) True
  - B) False
  - **Answer: B) False** (The time complexity is  $O(n \log n)$  only if the list is sorted, otherwise  $O(n^2)$  for brute force).
6. **The solution of the activity selection problem can have two overlapping activities in a particular time interval.**
- A) True

- B) False
  - **Answer: B) False** (The solution guarantees no overlapping activities).
7. **Consider the following number of activities with their start and finish time given below. Which of the following activity will be left out?**  
*(Note: Without the graph, I can't determine the correct activity. Please provide the graph if needed.)*
8. **Which among the following represents best-case time complexity for the activity selection problem?**
- A)  $O(n)$
  - B)  $O(1)$
  - C)  $O(n^2)$
  - D)  $O(n^2 * \log n)$
  - **Answer: A)  $O(n)$**  (Best case complexity when the activities are already sorted).
9. **Which of the following problems can be solved by a standard greedy algorithm?**
- I. Finding a minimum cost spanning tree in an undirected graph.
  - II. Finding a single maximal clique in a graph.
  - III. Finding the longest common subsequence in a given string.
  - A) I, II
  - B) I, II, III
  - C) III ONLY
  - D) I only
  - **Answer: D) I only** (A minimum cost spanning tree can be solved with a greedy algorithm, but the other problems require different approaches).
10. **The minimum number of colors with which we can guarantee that no two adjacent nodes will have the same color if we want to color the nodes of any cycle is:**
- A) 2
  - B) 3
  - C) 4
  - D)  $n+2$
  - **Answer: A) 2** (For any cycle, 2 colors are sufficient, as the cycle is bipartite).
1. **What is the minimum number of colors required to color a graph such that no two adjacent vertices have the same color called?**
- A) Chromatic number

- B) Chromatic index
  - C) Coloring number
  - D) Degree of the graph
  - **Answer: A) Chromatic number**
2. **In a graph, what is the condition for two vertices to be assigned different colors in a proper vertex coloring?**
- A) The vertices must have the same degree.
  - B) The vertices must be adjacent.
  - C) The vertices must be at an even distance from each other.
  - D) The vertices must belong to different connected components.
  - **Answer: B) The vertices must be adjacent.**
3. **Which of the following statements is true for a bipartite graph regarding vertex coloring?**
- A) It can be colored with three colors.
  - B) It can be colored with two colors.
  - C) It cannot be properly colored.
  - D) It requires more colors than the chromatic number.
  - **Answer: B) It can be colored with two colors.**
4. **What is the chromatic number of a complete graph  $K_n$  with  $n$  vertices?**
- A) 1
  - B) 2
  - C)  $n$
  - D)  $n - 1$
  - **Answer: C)  $n$**  (Each vertex must be assigned a different color in a complete graph, so the chromatic number is  $n$ ).
5. **Which algorithm is commonly used for finding an approximate solution to the vertex coloring problem in polynomial time?**
- A) Breadth-first search (BFS)
  - B) Depth-first search (DFS)
  - C) Greedy coloring algorithm
  - D) Dijkstra's algorithm
  - **Answer: C) Greedy coloring algorithm**



6. In edge coloring of a graph, what is the minimum number of colors required to color the edges such that no two adjacent edges share the same color called?
- A) Chromatic number
  - B) Chromatic index
  - C) Edge chromatic number
  - D) Line chromatic number
  - **Answer: C) Edge chromatic number**
7. What is the maximum number of colors required to color the edges of any simple graph  $G$  according to Vizing's theorem?
- A)  $\Delta(G) + 1$ , where  $\Delta(G)$  is the maximum degree of the graph
  - B)  $\Delta(G) - 1$
  - C)  $\lceil \Delta(G)/2 \rceil$
  - D)  $\Delta(G)$
  - **Answer: A)  $\Delta(G) + 1$**
8. What type of graph always requires at least three colors for proper coloring?
- A) Planar graph
  - B) Bipartite graph
  - C) Non-bipartite graph
  - D) Cycle graph with an odd number of vertices
  - **Answer: D) Cycle graph with an odd number of vertices**
9. What is the Four Color Theorem related to graph coloring?
- A) It states that any planar graph can be colored with at most four colors.
  - B) It states that any graph can be colored with four colors.
  - C) It states that a graph requires at least four colors if it contains a complete subgraph with four vertices.
  - D) It provides a method to color graphs using four different algorithms.
  - **Answer: A) It states that any planar graph can be colored with at most four colors.**
10. What is an NP-complete problem related to graph coloring?
- A) Determining the chromatic number of a graph
  - B) Finding a Hamiltonian cycle in a graph
  - C) Finding the shortest path between two vertices
  - D) Determining if a graph is bipartite

- **Answer: A) Determining the chromatic number of a graph**

❑ **What is the first step in the Huffman Coding algorithm?**

- A) Construct the Huffman Tree
- B) Count the frequency of each character
- C) Encode the characters
- D) Decode the characters
- **Answer: B) Count the frequency of each character**

❑ **Which of the following Java classes can be used to implement a min-heap for Huffman Coding?**

- A) Stack
- B) PriorityQueue
- C) ArrayList
- D) LinkedList
- **Answer: B) PriorityQueue**

❑ **When constructing the Huffman Tree, what happens to the two nodes with the lowest frequencies?**

- A) They are merged to form a new internal node
- B) They are discarded
- C) Their frequencies are swapped
- D) They are kept as leaf nodes
- **Answer: A) They are merged to form a new internal node**

❑ **What is the output of the Huffman Coding algorithm?**

- A) A binary search tree
- B) An encoded string
- C) A Huffman Tree and a table of codes
- D) A sorted array
- **Answer: C) A Huffman Tree and a table of codes**

❑ **In Huffman Coding, what is the property of the generated codes?**

- A) All codes have the same length
- B) Codes are prefix-free
- C) Codes are suffix-free
- D) Codes are generated randomly

- **Answer: B) Codes are prefix-free**

❑ **What is the main advantage of Huffman Coding over fixed-length coding?**

- A) Simplicity
- B) Speed
- C) Efficiency in space
- D) Consistency
- **Answer: C) Efficiency in space**

❑ **Which data structure is used to store the Huffman codes for each character?**

- A) Linked list
- B) Array
- C) HashMap
- D) Stack
- **Answer: C) HashMap**

❑ **What is the time complexity of building a Huffman Tree?**

- A)  $O(n)$
- B)  $O(n \log n)$
- C)  $O(n^2)$
- D)  $O(\log n)$
- **Answer: B)  $O(n \log n)$**

❑ **In Huffman Coding, how is the frequency of characters represented?**

- A) Using a binary search tree
- B) Using an array
- C) Using a min-heap
- D) Using a hash table
- **Answer: C) Using a min-heap**

❑ **What is the primary purpose of Huffman Coding?**

- A) Sorting elements in an array
- B) Data compression
- C) Searching for an element in a list
- D) Finding the shortest path in a graph
- **Answer: B) Data compression**

❓ In Java networking, which exception is commonly thrown for socket-related errors?

- A) IOException
- B) NullPointerException
- C) SQLException
- D) ClassNotFoundException
- **Answer: A) IOException**

❓ Which class is used to create a server-side socket in Java?

- A) ServerSocket
- B) Socket
- C) DatagramSocket
- D) InetAddress
- **Answer: A) ServerSocket**

❓ Which method is used to accept a connection from a client in the ServerSocket class?

- A) connect()
- B) bind()
- C) accept()
- D) listen()
- **Answer: C) accept()**

❓ Which class in Java is used for client-side socket programming?

- A) ServerSocket
- B) Socket
- C) DatagramSocket
- D) InetAddress
- **Answer: B) Socket**

❓ Which protocol is typically used with DatagramSocket in Java?

- A) TCP
- B) FTP
- C) HTTP
- D) UDP
- **Answer: D) UDP**

❓ Which class in Java provides methods to work with IP addresses?

- A) Socket
- B) ServerSocket
- C) InetAddress
- D) DatagramPacket
- **Answer: C) InetAddress**

❓ **Which method of the Socket class is used to get the input stream for reading data from the socket?**

- A) getOutputStream()
- B) getInputStream()
- C) connect()
- D) close()
- **Answer: B) getInputStream()**

❓ **Which method of the Socket class is used to send data to the socket?**

- A) getInputStream()
- B) getOutputStream()
- C) send()
- D) write()
- **Answer: B) getOutputStream()**

❓ **What is the default port number for HTTP?**

- A) 21
- B) 25
- C) 80
- D) 443
- **Answer: C) 80**

❓ **Which of the following is not a valid constructor for the DatagramPacket class?**

- A) DatagramPacket(byte[] buf, int length)
- B) DatagramPacket(byte[] buf, int offset, int length)
- C) DatagramPacket(byte[] buf, int length, InetAddress address, int port)
- D) DatagramPacket(byte[] buf, InetAddress address, int port)
- **Answer: D) DatagramPacket(byte[] buf, InetAddress address, int port)**

❓ **What is the primary purpose of encryption in Java applications?**

- a) To improve application performance
- b) To compress data for storage efficiency
- c) To protect sensitive data from unauthorized access
- d) To format data in a readable manner
- **Answer: c) To protect sensitive data from unauthorized access**

❑ Which Java class is commonly used for symmetric encryption?

- a) SecureRandom
- b) KeyPairGenerator
- c) Cipher
- d) MessageDigest
- **Answer: c) Cipher**

❑ Which algorithm is an example of asymmetric encryption?

- a) AES
- b) DES
- c) RSA
- d) Blowfish
- **Answer: c) RSA**

❑ What does the term “symmetric encryption” refer to?

- a) Using the same key for both encryption and decryption
- b) Using different keys for encryption and decryption
- c) Using a public key for encryption and a private key for decryption
- d) Encrypting data without the need for a key
- **Answer: a) Using the same key for both encryption and decryption**

❑ Which Java library provides cryptographic operations, including encryption and decryption?

- a) java.util
- b) java.io
- c) java.security
- d) java.net
- **Answer: c) java.security**

❑ What is the role of the SecretKey interface in Java?

- a) To generate random numbers

- b) To create message digests
- c) To represent a secret (symmetric) key used in encryption
- d) To manage key pairs for asymmetric encryption
- **Answer: c) To represent a secret (symmetric) key used in encryption**

❓ **Which method of the Cipher class initializes the cipher for encryption or decryption?**

- a) init()
- b) start()
- c) setup()
- d) initialize()
- **Answer: a) init()**

❓ **How can you convert a byte array to a hexadecimal string in Java?**

- a) Using the Base64 class
- b) Using the Hex class
- c) Using the Arrays class
- d) Using the DatagramPacket class
- **Answer: b) Using the Hex class**

❓ **What is a common block cipher mode of operation used with the AES algorithm in Java?**

- a) ECB (Electronic Codebook)
- b) CBC (Cipher Block Chaining)
- c) OFB (Output Feedback)
- d) CFB (Cipher Feedback)
- **Answer: b) CBC (Cipher Block Chaining)**

❓ **Which method of the Cipher class is used to perform the actual encryption or decryption?**

- a) doFinal()
- b) update()
- c) process()
- d) execute()
- **Answer: a) doFinal()**

