

Chapter 11: The World of Algorithms - Practice Question Bank

Easy Level

Multiple Choice Questions (MCQs)

1. What is an algorithm in mathematics and computer science?
 - A) A random guess to solve a problem
 - B) A step-by-step procedure or systematic procedure to arrive at an answer
 - C) A geometric shape
 - D) A type of prime number
2. In the addition algorithm, how are numbers initially written down in the Indian base-ten place-value system?
 - A) Aligned from left to right
 - B) Aligned from right to left
 - C) Placed side by side randomly
 - D) Arranged in ascending order
3. What is the value of carry when adding two digits if their sum is less than 10?
 - A) 1
 - B) 0
 - C) 9
 - D) 10
4. What is the value of carry when adding two digits if their sum is 10 or more?
 - A) 0
 - B) 1
 - C) 2
 - D) 10
5. What does gcd stand for?
 - A) Greatest Common Denominator
 - B) Greatest Common Divisor
 - C) General Common Divisor

- D) Greatest Composite Divisor
6. What is another commonly used term for Greatest Common Divisor (gcd)?
- A) Least Common Multiple (lcm)
 - B) Highest Common Factor (hcf)
 - C) Prime Factorization
 - D) Remainder
7. What is initially the list of divisors when starting the algorithm to find divisors of a number n ?
- A) [1]
 - B) [1, n]
 - C) An empty list []
 - D) [0]
8. In the divisor-finding algorithm for a number n , which numbers are checked as possible divisors of n ?
- A) From 1 down to 0
 - B) Each number j in the sequence 1, 2, 3, ..., n
 - C) Only even numbers
 - D) Only prime numbers
9. Which number divides every natural number without leaving a remainder?
- A) 0
 - B) 1
 - C) 2
 - D) n itself
10. What is the list of divisors of 18 obtained in ascending order?
- A) [1, 2, 3, 6, 9, 18]
 - B) [2, 3, 6, 9]
 - C) [1, 2, 9, 18]
 - D) [1, 3, 6, 18]
11. In Euclid's original subtraction algorithm, $\text{gcd}(m, n)$ for $m \geq n$ is shown to be equal to:
- A) $\text{gcd}(m, n + m)$
 - B) $\text{gcd}(n, m - n)$
 - C) $\text{gcd}(m - n, n + m)$

- D) $\gcd(m, m \bmod n)$

12. In Euclid's algorithm, what is the stopping condition or base case when computing $\gcd(m, 0)$?

- A) 0
- B) 1
- C) m
- D) n

13. What is $17 \bmod 5$?

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

14. What is $33 \bmod 7$?

- A) 3
- B) 4
- C) 5
- D) 6

15. Who was the influential Persian mathematician whose work on Indian numerals and arithmetic led to the word "algorithm"?

- A) Aryabhata
- B) Euclid
- C) Al-Khwarizmi
- D) Brahmagupta

16. What was the Latin title of Al-Khwarizmi's influential 12th-century translated book on Indian numerals?

- A) Elements
- B) Aryabhatiya
- C) Liber Algorismi de numero Indorum
- D) Vṛttajāṭisamuchaya

17. A list is an example of what computer science concept mentioned in the text?

- A) A data structure
- B) A prime factor

- C) A carry-over digit
- D) A remainder

18. What is the value of $\gcd(23, 23)$?

- A) 0
- B) 1
- C) 23
- D) 46

19. In the improved gcd algorithm (Aryabhata's division algorithm), the reduction step replaces $m - n$ with:

- A) $m * n$
- B) $m \bmod n$
- C) $m + n$
- D) m / n

20. What is the primary advantage of algorithms over explicit counting (like drawing and counting dots)?

- A) They use more paper
- B) They are much faster and scale better with large numbers
- C) They require no assumptions
- D) They eliminate the need for numbers

Easy Level MCQ Answer Key

1. B
2. B
3. B
4. B
5. B
6. B
7. C
8. B
9. B
10. A
11. B
12. C
13. B
14. C
15. C
16. C
17. A

- 18. C
- 19. B
- 20. B

Easy Level Subjective Questions

1. Define what an algorithm is in your own words based on the text.
2. Why is aligning numbers from right to left crucial in the standard addition algorithm?
3. What is the role of the carry digit in the addition algorithm, and what is its maximum possible value when adding two single digits with a carry?
4. What basic assumption is made when using the addition algorithm?
5. How does explicit dot-counting compare in terms of time to the step-by-step addition algorithm as numbers grow larger?
6. Define the greatest common divisor (gcd) of two numbers.
7. Describe the first step in finding the divisors of a number n using the algorithm provided in the text.
8. What is the output list of divisors of the number 18 when running the divisor algorithm?
9. Why are the divisors generated by the systematic check from 1 to n naturally in ascending order?
10. Explain what a data structure is with reference to lists.
11. State Euclid's basic observation regarding $\gcd(m, n)$ when $m \geq n$ using subtraction.
12. What stopping condition signals the end of Euclid's gcd subtraction/division algorithm?
13. Calculate $375 \bmod 100$.
14. Calculate $825 \bmod 75$.
15. Explain how the word "algorithm" etymologically originated from Al-Khwarizmi's name and Indian arithmetic.
16. What is the main difference between Euclid's original subtraction algorithm and Aryabhata's division algorithm?
17. Why is Aryabhata's division algorithm more efficient than Euclid's subtraction algorithm for numbers like $\gcd(99, 2)$?
18. If you have an empty list of common divisors initially, what number is guaranteed to be a common divisor for any two positive integers?
19. What does executing an algorithm step-by-step mean?
20. Give an example of a stopping condition in an algorithm.

Easy Level Subjective Answers

1. An algorithm is a precise, step-by-step systematic procedure or set of rules designed to solve a specific problem or class of problems.
2. Aligning columns from right to left ensures that digits of the same place value (units, tens, hundreds, etc.) are added together correctly.
3. The carry digit represents a newly formed group of ten that is carried over to the next place value on the left. Its maximum value when adding two single-digit numbers plus a previous carry of 1 is 1 (since $9 + 9 + 1 = 19$, where 1 is carried).

4. The addition algorithm assumes that we already know how to add single-digit numbers directly without recounting them.
 5. Explicit dot-counting takes time proportional to the value of the numbers (requiring 10 times or 100 times more dots for larger numbers), whereas the algorithm takes time proportional to the number of digits (columns), making it vastly faster.
 6. The greatest common divisor (gcd) of two numbers is the largest positive integer that divides both numbers without leaving a remainder.
 7. The algorithm starts with an empty list-of-divisors (denoted as []).
 8. $\text{divisors}(18) = [1, 2, 3, 6, 9, 18]$.
 9. They are in ascending order because the algorithm checks possible divisors systematically from 1 up to n .
 10. A data structure is a way of organising information or data (such as lists of numbers) to help make an algorithm more efficient.
 11. Euclid observed that if $m \geq n$, then $\text{gcd}(m, n) = \text{gcd}(n, m - n)$.
 12. The algorithm stops when the second argument reaches 0 (i.e., $\text{gcd}(m, 0)$), at which point the answer is m .
 13. $375 \bmod 100 = 75$.
 14. $825 \bmod 75 = 0$.
 15. The Latin translation of Al-Khwarizmi's book on Indian numerals was titled "Liber Algorismi de numero Indorum", where "Algorismi" referred to Al-Khwarizmi. This evolved into "algorism" and later "algorithm", originally denoting Indian arithmetic methods before broadening to mean any systematic step-by-step procedure.
 16. Euclid's original algorithm uses repeated subtraction (reducing $\text{gcd}(m, n)$ to $\text{gcd}(n, m - n)$), whereas Aryabhata's division algorithm uses repeated division by remainder (reducing $\text{gcd}(m, n)$ to $\text{gcd}(n, m \bmod n)$).
 17. Aryabhata's division algorithm uses remainders (long division), which reduces numbers in logarithmic/digit-proportional steps rather than subtracting small numbers one by one, requiring only 2 steps for $\text{gcd}(99, 2)$ instead of ~49 steps.
 18. The number 1 is guaranteed to be a common divisor for any two positive integers.
 19. Executing an algorithm step-by-step means running through its defined instructions sequentially, updating intermediate variables and lists until completion.
 20. An example of a stopping condition is when n becomes 0 in the gcd algorithm, or when all digits have been processed in the addition algorithm.
-

Moderate Level

Multiple Choice Questions (MCQs)

1. In the algorithm to find divisors of n , how many numbers are checked in total between 1 and n ?
 - A) $n/2$ numbers

- B) n numbers
 - C) $\sqrt{n}$ numbers
 - D) $\log(n)$ numbers
2. When combining divisor scans for m and n up to $\max(m, n)$, how many potential divisor checks are performed?
- A) $m + n$
 - B) $\max(m, n)$
 - C) $\min(m, n)$
 - D) $m * n$
3. If we optimize the common divisor search by checking up to the smaller of the two numbers, what is the upper limit of the scan?
- A) $\max(m, n)$
 - B) $\min(m, n)$
 - C) $m + n$
 - D) $m * n$
4. In the algorithm that uses `most-recent-common-divisor`, what initial value is assigned to it before scanning?
- A) 0
 - B) 1
 - C) $\min(m, n)$
 - D) $\max(m, n)$
5. What is `divisors(375)` as given in the text?
- A) [1, 3, 5, 15, 25, 75, 125, 375]
 - B) [1, 3, 5, 11, 15, 25, 33, 55, 75, 165, 275, 825]
 - C) [1, 5, 25, 125, 375]
 - D) [1, 3, 15, 75, 375]
6. What is `divisors(825)` as given in the text?
- A) [1, 3, 5, 15, 25, 75, 125, 375]
 - B) [1, 3, 5, 11, 15, 25, 33, 55, 75, 165, 275, 825]
 - C) [1, 11, 33, 55, 165, 275, 825]
 - D) [1, 3, 25, 75, 825]
7. What is `gcd(375, 825)`?

- A) 25
- B) 75
- C) 125
- D) 15

8. What is $\gcd(54000, 81000)$?

- A) 9000
- B) 13500
- C) 27000
- D) 54000

9. In Euclid's subtraction algorithm, if $m = 825$ and $n = 375$, what is the first reduction step?

- A) $\gcd(375, 450)$
- B) $\gcd(825, 450)$
- C) $\gcd(375, 375)$
- D) $\gcd(825, 375)$

10. How many reduction steps does Euclid's subtraction algorithm take for $\gcd(99, 2)$?

- A) 2 steps
- B) 10 steps
- C) About 49 steps
- D) 99 steps

11. How many reduction steps does Aryabhata's division algorithm take for $\gcd(99, 2)$?

- A) 1 step
- B) 2 steps
- C) 49 steps
- D) 99 steps

12. In the long division execution of $\gcd(60, 16)$, what is the first remainder obtained when 60 is divided by 16?

- A) 0
- B) 4
- C) 12
- D) 16

13. What is the final gcd result when computing $\gcd(60, 16)$ via Aryabhata's algorithm?

- A) 2

- B) 4
- C) 8
- D) 12

14. If $m < n$ in Aryabhata's division algorithm, what is the first action taken?

- A) Stop and report error
- B) Reverse the numbers and compute $\gcd(n, m)$
- C) Compute $m \bmod n$
- D) Subtract m from n

15. Why does the number of steps in Aryabhata's division algorithm scale much better than Euclid's subtraction algorithm?

- A) It uses addition instead of subtraction
- B) The number of reductions is proportional to the number of digits rather than the magnitude of the values
- C) It ignores remainders
- D) It only works for even numbers

16. What is the primary purpose of assigning names to intermediate quantities (like `list-of-divisors` or `most-recent-common-divisor`)?

- A) To make the algorithm run slower
- B) For writing down algorithms concisely and keeping track of values
- C) To encrypt the data
- D) To eliminate variables

17. Divisors naturally occur in pairs (e.g., for 18, pairs are (1,18), (2,9), (3,6)). How many numbers do we need to examine up to to find all divisors if exploiting this property?

- A) Up to n
- B) Up to $n/2$
- C) Up to $\sqrt{n}$
- D) Up to $\log(n)$

18. If List 1 and List 2 are two lists in increasing order, finding elements in List 1 not present in List 2 is an exercise in:

- A) Data structure manipulation and set operations
- B) Matrix multiplication
- C) Prime factorization
- D) Random sampling

19. When computing divisors of n by examining numbers from n down to 1 (reverse order), how does the resulting list order change compared to checking 1 to n ?

- A) The list is in descending order
- B) The list remains in ascending order
- C) The list becomes random
- D) The algorithm fails completely

20. What is $51000 \bmod 81000$ (or vice versa in gcd context)?

- A) 30000
- B) 51000
- C) 81000
- D) 27000

Moderate Level MCQ Answer Key

1. B
2. B
3. B
4. B
5. A
6. B
7. B
8. C
9. A
10. C
11. B
12. B
13. B
14. B
15. B
16. B
17. C
18. A
19. A
20. A

Moderate Level Subjective Questions

1. Write out the step-by-step execution of the algorithm to find divisors(15).
2. Trace the execution of the **most-recent-common-divisor** algorithm for $m = 6$ and $n = 12$.
3. Using Euclid's subtraction algorithm, trace the first three reduction steps for $\gcd(825, 375)$.

4. Using Aryabhata's division algorithm (long division), compute $\text{gcd}(60, 16)$ showing each intermediate remainder and reduction.
5. Explain why checking up to $\min(m, n)$ is sufficient when searching for common divisors of m and n .
6. How does doing away with lists (using `most-recent-common-divisor`) affect memory and execution storage during a gcd scan?
7. Explain the mathematical justification behind reducing $\text{gcd}(m, n)$ to $\text{gcd}(n, m - n)$.
8. Explain the mathematical justification behind reducing $\text{gcd}(m, n)$ to $\text{gcd}(n, m \bmod n)$ using the concept of 'tiling'.
9. Why does Euclid's subtraction algorithm require approximately k steps for $\text{gcd}(2k + 1, 2)$?
10. What is the complexity growth factor of the naive divisor/gcd scan algorithms when the number of digits increases from 3 to 4 (i.e., when values increase 10-fold)?
11. Describe how divisors occur in pairs and how this observation can optimize divisor finding up to $\sqrt{n}$.
12. Outline the algorithm to check if List 1 elements are not present in List 2 (assuming both are sorted in increasing order).
13. What happens if you attempt to add a 5-digit number and a 3-digit number using the standard addition algorithm? Does it work correctly?
14. Why is step 5 (handling final carry 1) necessary in the addition algorithm? Give an example where omitting it fails.
15. Outline an algorithm `prime(n)` to check if a number n is prime using the `divisors(n)` function.
16. Outline an algorithm `primedivisors(n)` to compute the list of prime divisors of n .
17. How does Al-Khwarizmi's Latin treatise *Liber Algorithmi de numero Indorum* connect Indian mathematics to modern computing terminology?
18. Trace what happens in the divisor algorithm if we examine numbers from n down to 1 instead of 1 up to n .
19. If we work backwards from $\min(m, n)$ down to 1 in the common divisor search, how does that affect when we find the gcd?
20. Discuss why justifying an algorithm's correctness is just as important as writing down its steps.

Moderate Level Subjective Answers

1. `divisors(15)`: Start with []. Check 1 (divides $\rightarrow$ [1]), 2 (no), 3 (divides $\rightarrow$ [1,3]), 4 (no), 5 (divides $\rightarrow$ [1,3,5]), 6 to 14 (no), 15 (divides $\rightarrow$ [1,3,5,15]). Result: [1,3,5,15].
2. $m=6, n=12$. $\min(6,12)=6$. Start `most-recent-common-divisor` = 1. $k=2$: divides both $\rightarrow$ update to 2. $k=3$: divides both $\rightarrow$ update to 3. $k=4$: does not divide 6 $\rightarrow$ remains 3. $k=5$: does not divide $\rightarrow$ remains 3. $k=6$: divides both $\rightarrow$ update to 6. Final gcd = 6.
3. $\text{gcd}(825, 375)$: Step 1: $375 < 825$, reverse $\rightarrow \text{gcd}(825, 375)$. Step 3: reduce to $\text{gcd}(375, 825 - 375) = \text{gcd}(375, 450)$. Next: $375 < 450$, reverse $\rightarrow \text{gcd}(450, 375)$. Next reduce: $\text{gcd}(375, 450 - 375) = \text{gcd}(375, 75)$.

4. $\text{gcd}(60, 16)$: $60 = 16 \cdot 3 + 12 \rightarrow \text{gcd}(16, 12)$. $16 = 12 \cdot 1 + 4 \rightarrow \text{gcd}(12, 4)$. $12 = 4 \cdot 3 + 0 \rightarrow \text{gcd}(4, 0)$. Result = 4.
5. A common divisor of two numbers cannot exceed the smaller of the two numbers because any divisor of a number must be less than or equal to that number. Thus, checking beyond $\min(m, n)$ is redundant.
6. Doing away with lists avoids storing all intermediate common divisors in memory; we only overwrite a single integer variable (**most-recent-common-divisor**), drastically saving memory and storage overhead.
7. If d divides both m and n , then $m = ad$ and $n = bd$. Then $m - n = ad - bd = (a - b)d$, so d divides $m - n$. Conversely, if d divides n and $m - n$, then $m = n + (m - n) = xd + yd = (x + y)d$, so d divides m . Thus the set of common divisors is identical.
8. A number d 'tiles' both m and n if it divides both without remainder. Since m divided by n leaves remainder $r = m \bmod n$ (meaning $m = q \cdot n + r$), any number d that tiles n and m will also tile r ($m \bmod n$), and vice versa.
9. For $\text{gcd}(2k+1, 2)$, subtracting 2 repeatedly from $2k+1$ takes k subtraction steps before reaching 1 and 2, making the step count proportional to the magnitude k rather than the number of digits.
10. The work grows by a factor of 10 because the naive algorithm scans all numbers up to the value (or $\min(m, n)$), so a 10-fold increase in number magnitude requires 10 times as many scan steps.
11. Divisors come in pairs (a, b) such that $a \cdot b = n$. Since one factor in a pair must be less than or equal to $\sqrt{n}$, we only need to check divisors up to $\sqrt{n}$ to find all divisor pairs.
12. Iterate through each element in sorted List 1; check if it exists in sorted List 2 (using pointers or binary search); if not found, add it to a new result list.
13. Yes, it works correctly as long as place values are aligned on the right. Shorter numbers effectively have leading implicit zeros in higher columns.
14. Step 5 (writing 1 to the left if final carry is 1) is necessary when the leftmost column addition generates a carry. Example: $7 + 5 = 12$; if you drop the carry 1, you get 2 instead of 12.
15. **prime(n)**: Compute **divisors(n)**. If the resulting list has length exactly 2 (i.e., $[1, n]$), then n is prime; otherwise, it is not prime.
16. **primedivisors(n)**: Compute **divisors(n)**, then filter the list to retain only those numbers that are prime (using a primality test on each element).
17. Al-Khwarizmi's name in Latin ("Algorismi") gave rise to "algorism" and "algorithm", originally describing Indian arithmetic methods (place-value, addition, multiplication, division algorithms) that formed the foundation of computational procedures.
18. If we check from n down to 1, the resulting list of divisors will be generated in descending order (largest to smallest) rather than ascending order.
19. Working backwards from $\min(m, n)$ down to 1 means the very first common divisor you encounter is guaranteed to be the greatest common divisor (gcd), allowing you to stop immediately upon finding the first match.
20. Justifying an algorithm guarantees its correctness and ensures it produces accurate results across all valid inputs, preventing logical flaws or edge-case failures.

Difficult Level

Multiple Choice Questions (MCQs)

1. Using the improved version of Euclid's algorithm, what is $\text{gcd}(375, 825)$?
 - A) 25
 - B) 75
 - C) 125
 - D) 375
2. Using the improved version of Euclid's algorithm, what is $\text{gcd}(51000, 81000)$?
 - A) 3000
 - B) 9000
 - C) 17000
 - D) 51000
3. In the prime checking algorithm `prime(n)`, how many distinct factors must a prime number p have?
 - A) Exactly 1
 - B) Exactly 2 (1 and p)
 - C) Any prime number of factors
 - D) More than 2
4. When computing the prime factorization of 180, which is represented as $2^2 \times 3^2 \times 5^1$, how is this data typically structured in an algorithm?
 - A) As a single integer 180
 - B) As a list of prime-exponent pairs or dictionary/mapping
 - C) As a list of all individual divisors
 - D) As a sum of digits
5. If $m \geq n$, what formal logic proves that d divides both m and n if and only if d divides both n and $(m \bmod n)$?
 - A) Division theorem / Euclidean division relation $m = qn + r$
 - B) Prime number theorem
 - C) Quadratic reciprocity
 - D) Binomial theorem

6. What is the primary theoretical advantage of Aryabhata's division algorithm over Euclid's subtraction algorithm in terms of computational complexity?
- A) It runs in time proportional to the value of the numbers
 - B) It runs in time proportional to the number of digits (logarithmic in value)
 - C) It requires no memory
 - D) It uses addition instead of division
7. In the analysis of addition algorithms, if counting 1 dot per second takes 1 hour 40 minutes for $347 + 2729$, approximately how long does the single-digit addition algorithm take?
- A) About 1 minute
 - B) Exactly 1 hour 40 minutes
 - C) 10 hours
 - D) 5 seconds
8. If we modify the divisor-finding algorithm to check numbers from n down to 1, what is the time complexity in terms of basic division checks?
- A) $O(\log n)$
 - B) $O(\sqrt{n})$
 - C) $O(n)$
 - D) $O(n^2)$
9. When comparing prime factorizations of two numbers $m = p_1^{a_1} p_2^{a_2} \dots$ and $n = p_1^{b_1} p_2^{b_2} \dots$, how is their gcd determined from their exponents?
- A) Take the sum of exponents $a_i + b_i$
 - B) Take the minimum of exponents $\min(a_i, b_i)$ for each prime factor
 - C) Take the maximum of exponents $\max(a_i, b_i)$ for each prime factor
 - D) Multiply exponents $a_i * b_i$
10. Similarly, how is the least common multiple (lcm) of m and n determined from their prime factorization exponents?
- A) $\min(a_i, b_i)$
 - B) $\max(a_i, b_i)$
 - C) $a_i + b_i$
 - D) $|a_i - b_i|$
11. What is the output of `primedivisors(18)`?

- A) [1, 2, 3, 6, 9, 18]
- B) [2, 3]
- C) [1, 2, 3]
- D) [2, 3, 18]

12. In the context of adding decimal fractions, how must the addition algorithm be modified regarding alignment?

- A) Align from left to right
- B) Align the decimal points directly above one another
- C) Ignore decimal points entirely
- D) Pad with letters

13. What happens if you execute the subtraction-based gcd algorithm on two very large prime numbers that are close to each other, such as $\text{gcd}(1000003, 1000001)$?

- A) It finishes in 1 step
- B) It takes millions of subtraction reduction steps
- C) It results in a division by zero
- D) It fails to terminate

14. How many reduction steps does Aryabhata's division algorithm take for $\text{gcd}(1000003, 1000001)$?

- A) 2 steps (since $1000003 \bmod 1000001 = 2$, then $1000001 \bmod 2 = 1$, then $2 \bmod 1 = 0$)
- B) 1000001 steps
- C) 500000 steps
- D) 0 steps

15. In the House of Wisdom in Baghdad during the 8th-9th century, which mathematician translated and disseminated Indian arithmetic?

- A) Euclid
- B) Al-Khwarizmi
- C) Aryabhata
- D) Fibonacci

16. What does the notation $m \bmod n$ formally return?

- A) The quotient when m is divided by n
- B) The integer remainder r such that $m = qn + r$ with $0 \leq r < n$
- C) The fraction part m / n
- D) The greatest common divisor

17. If an algorithm takes $m + n$ steps in its combined scan, and we optimize it to $\max(m, n)$, and further to $\min(m, n)$, what asymptotic complexity class do all these linear scans belong to?

- A) $O(1)$ constant
- B) $O(\log n)$ logarithmic
- C) $O(n)$ linear
- D) $O(n^2)$ quadratic

18. Why can't the naive divisor-finding algorithm be considered an "efficient" algorithm for numbers with 20 or 30 digits?

- A) Because computers cannot store numbers that large
- B) Because the number of operations is proportional to the value of the number, which is astronomically huge (10^{20}), exceeding the age of the universe in operations
- C) Because division is mathematically impossible for large numbers
- D) Because lists are not data structures

19. In Exercise Set 11.2, computing elements in List 1 not present in List 2 assumes the lists are:

- A) Unsorted and randomized
- B) In increasing order
- C) Of equal length
- D) Composed only of prime numbers

20. What is the LCM of two numbers m and n related to their gcd by the standard formula?

- A) $\text{lcm}(m, n) = m + n - \text{gcd}(m, n)$
- B) $\text{lcm}(m, n) = \frac{m \times n}{\text{gcd}(m, n)}$
- C) $\text{lcm}(m, n) = \text{gcd}(m, n)^2$
- D) $\text{lcm}(m, n) = m \times n \times \text{gcd}(m, n)$

Difficult Level MCQ Answer Key

1. B
2. B
3. B
4. B
5. A
6. B
7. A
8. C

9. B
10. B
11. B
12. B
13. B
14. A
15. B
16. B
17. C
18. B
19. B
20. B

Difficult Level Subjective Questions

1. Formally prove using the division theorem ($m = qn + r$) that $\text{gcd}(m, n) = \text{gcd}(n, m \bmod n)$.
2. Design the complete algorithm `prime(n)` making use of `divisors(n)`.
3. Design the algorithm `primedivisors(n)` that outputs the list of prime divisors of n .
4. Explain how to compute the prime factorization of a number algorithmically and how to represent it (e.g., $180 = 2^2 \times 3^2 \times 5^1$).
5. Using prime factorization exponents, write out the algorithm to compute $\text{gcd}(m, n)$ and $\text{lcm}(m, n)$.
6. Compare the execution steps and efficiency of Euclid's subtraction algorithm versus Aryabhata's division algorithm for $\text{gcd}(1000003, 1000001)$.
7. Modify the standard addition algorithm to correctly add two decimal fractions with differing numbers of decimal places.
8. Write an algorithm to find elements present in List 1 that are not present in List 2, assuming both lists are sorted in increasing order (linear merge-style check).
9. Explain why the number of reduction steps in Aryabhata's division algorithm is proportional to the number of digits rather than the magnitude of the numbers.
10. Detail the historical journey of the word "algorithm" from Al-Khwarizmi's Latinized name to its modern definition.
11. Analyze the time complexity of the `most-recent-common-divisor` algorithm and explain why it is still $O(\min(m, n))$ rather than $O(\log(\min(m, n)))$.
12. How would you modify the divisor-finding algorithm to exploit divisor pairs (a, b) where $a \times b = n$, reducing the checks to $\sqrt{n}$?
13. Write an algorithm to compute the least common multiple (lcm) of two numbers using the gcd algorithm.
14. Discuss the concept of invariants in algorithms and how they help justify correctness (especially in gcd algorithms).
15. Explain why the size of input (number of digits vs. value of number) is a critical metric when analyzing algorithm efficiency.

16. Trace the long division steps of Aryabhata's algorithm for $\text{gcd}(1789287, 237656)$.
17. What happens in the divisor algorithm if $n = 1$? Trace the execution steps.
18. Explain how data structures like sorted lists improve algorithmic efficiency compared to unstructured collections.
19. If an algorithm has a time complexity proportional to $O(n)$, and n increases from 1,000 to 1,000,000, by what factor does the execution time increase?
20. Summarize the core contributions of Aryabhata and Euclid to the computational science of greatest common divisors.

Difficult Level Subjective Answers

1. Let $m = qn + r$, where $r = m \bmod n$. Any common divisor d of m and n must divide $m - qn = r$, so d divides n and r . Conversely, any common divisor d of n and r must divide $qn + r = m$, so d divides m and n . Thus the sets of common divisors are identical, proving $\text{gcd}(m, n) = \text{gcd}(n, r)$.
2. `prime(n)`: 1. Let `div = divisors(n)`. 2. If length of `div` equals 2, return True. 3. Else return False.
3. `primedivisors(n)`: 1. Let `div = divisors(n)`. 2. Initialize `prime-div-list = []`. 3. For each `x` in `div`: if `prime(x)` is True, add `x` to `prime-div-list`. 4. Return `prime-div-list`.
4. Prime factorization algorithm: Trial division. Divide n by 2 repeatedly while divisible, recording exponent; then trial divide by odd numbers 3, 5, 7, ... up to $\sqrt{n}$. Represent as a list of pairs `[(2, 2), (3, 2), (5, 1)]` or dictionary `{2: 2, 3: 2, 5: 1}`.
5. For prime factorizations $m = \prod p_i^{a_i}$ and $n = \prod p_i^{b_i}$: $\text{gcd}(m, n) = \prod p_i^{\min(a_i, b_i)}$ and $\text{lcm}(m, n) = \prod p_i^{\max(a_i, b_i)}$.
6. For $\text{gcd}(1000003, 1000001)$ (two primes close to each other): Euclid's subtraction takes 1 million steps ($1000003 - 1000001 = 2$, then 500,000 subtractions of 2). Aryabhata's division takes 3 steps: $1000003 \bmod 1000001 = 2$, then $1000001 \bmod 2 = 1$, then $2 \bmod 1 = 0$, finishing instantly.
7. Decimal addition algorithm: 1. Pad the shorter decimal with trailing zeros so both have the same number of decimal places. 2. Align decimal points vertically. 3. Perform integer addition from right to left across all columns. 4. Place the decimal point in the result directly aligned with the addends.
8. `list1_minus_list2(List1, List2)`: Use two pointers `i = 0, j = 0`. While `i < length(List1)` and `j < length(List2)`: if `List1[i] < List2[j]`, add `List1[i]` to result and `i++`; else if `List1[i] > List2[j]`, `j++`; else `i++, j++`. Add remaining elements of `List1` to result.
9. Due to the properties of Euclidean division and the Fibonacci worst-case bound (Lamé's theorem), the number of division steps is bounded by 5 times the number of digits in base 10, making it logarithmic in magnitude (linear in digits).

10. Al-Khwarizmi studied Indian mathematical texts in Baghdad (~820 CE). His treatise was translated into Latin as *Liber Algorithmi de numero Indorum*. "Algorithmi" became "algorithm" (Indian arithmetic methods) and later "algorithm" for any systematic step-by-step procedure.
 11. The **most-recent-common-divisor** algorithm still iterates through every integer from 2 up to $\min(m, n)$ in the worst case (e.g., when numbers are coprime or share small divisors), performing a linear scan of size $\min(m, n)$, which is $O(\min(m, n))$, not logarithmic.
 12. Iterate j from 1 up to $\lfloor \sqrt{n} \rfloor$. If j divides n , add j and n/j to the divisor list, then sort and remove duplicates.
 13. **lcm(m, n)**: Compute $g = \text{gcd}(m, n)$. Return $(m \times n) / g$.
 14. An invariant is a condition that remains true throughout the execution of an algorithm (e.g., the gcd of the working pair remains constant at every reduction step). Invariants ensure that when the algorithm terminates, the output is mathematically correct.
 15. Input size measured in digits determines scalability. An algorithm operating on value takes exponential time relative to digits (since value = 10^{digits}), whereas digit-proportional algorithms scale polynomially/logarithmically.
 16. $\text{gcd}(1789287, 237656)$: $1789287 = 7 \times 237656 + 125695 \rightarrow \text{gcd}(237656, 125695)$. $237656 = 1 \times 125695 + 111961 \rightarrow \text{gcd}(125695, 111961)$, and so on until remainder 0.
 17. If $n = 1$, **divisors(1)** checks j from 1 to 1. 1 divides 1, list becomes **[1]**. Divisor algorithm completes instantly with **[1]**.
 18. Sorted lists allow binary search or linear two-pointer traversal ($O(n)$ time) instead of nested loops ($O(n^2)$ time), dramatically speeding up comparisons, membership checks, and set operations.
 19. The execution time increases by a factor of 1,000 (from 1,000 to 1,000,000 steps), since time scales linearly ($O(n)$) with input magnitude.
 20. Euclid systematized foundational deductive geometry and early subtraction-based divisibility theory in *Elements*, while Aryabhata and Indian mathematicians introduced positional place-value numerals and efficient long-division remainder algorithms that revolutionized computational mathematics.
-

Olympiad Level

Multiple Choice Questions (MCQs)

1. By Lamé's Theorem, the number of division steps N required in the Euclidean algorithm for two positive integers $a > b$ with $b < 10^k$ is at most:
 - A) k
 - B) $5k$

- C) $10k$
 - D) k^2
2. If a and b are two positive integers such that $\text{gcd}(a, b) = 1$, what is $\text{gcd}(a + b, a - b)$?
- A) Always 1
 - B) Always 1 or 2
 - C) Always $a \times b$
 - D) Always prime
3. What is the asymptotic time complexity of computing $\text{gcd}(m, n)$ using Aryabhata's division algorithm where $m, n < N$, expressed in terms of the bit length $b = \log_2 N$?
- A) $O(2^b)$
 - B) $O(b)$
 - C) $O(b^2)$ (or $O(b^3)$ with naive division)
 - D) $O(1)$
4. If $\text{lcm}(a, b, c) = L$, and $\text{gcd}(a, b) = d_1$, $\text{gcd}(b, c) = d_2$, $\text{gcd}(c, a) = d_3$, which of the following represents a property of triple gcd/lcm relations?
- A) $\text{lcm}(a, b) \times \text{lcm}(b, c) \times \text{lcm}(c, a) = a b c \times \text{gcd}(a, b, c)$
 - B) $\text{lcm}(a, b, c) = \frac{abc \cdot \text{gcd}(a, b, c)}{\text{gcd}(a, b) \times \text{gcd}(b, c) \times \text{gcd}(c, a)}$
 - C) $\text{lcm}(a, b, c) = abc$ always
 - D) $\text{lcm}(a, b, c) = \text{gcd}(a, b, c)^3$
5. In the study of algorithms, what does the recurrence relation $T(n) = T(n/2) + O(1)$ solve to?
- A) $O(n)$
 - B) $O(n \log n)$
 - C) $O(\log n)$
 - D) $O(n^2)$
6. If F_n denotes the n -th Fibonacci number ($F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, \dots$), for which input pair (a, b) does the Euclidean algorithm take the maximum number of division steps for a given number of digits (Worst-Case Inputs)?
- A) (F_{n+1}, F_n)

- B) $(2^n, 3^n)$
 - C) $(n, n-1)$
 - D) $(10^n, 1)$
7. What is the value of $\sum_{k=1}^n \phi(k)$ where $\phi(k)$ is Euler's totient function, in terms of asymptotic growth?
- A) $O(n)$
 - B) $O(n \log n)$
 - C) $O(n^2)$
 - D) $O(\sqrt{n})$
8. If $d = \text{gcd}(a, b)$, Bézout's Identity states that there exist integers x and y such that:
- A) $ax + by = d$
 - B) $ax - by = d^2$
 - C) $a^x + b^y = d$
 - D) $x/a + y/b = d$
9. Which algorithm computes the coefficients x and y in Bézout's Identity $ax + by = \text{gcd}(a, b)$?
- A) Sieve of Eratosthenes
 - B) Extended Euclidean Algorithm
 - C) Newton-Raphson Method
 - D) Binary Search Algorithm
10. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, which of the following modular arithmetic properties is FALSE?
- A) $(a + c) \equiv (b + d) \pmod{m}$
 - B) $(a - c) \equiv (b - d) \pmod{m}$
 - C) $(a \times c) \equiv (b \times d) \pmod{m}$
 - D) $(a / c) \equiv (b / d) \pmod{m}$ (always without restrictions)
11. What is the number of divisors function $d(n)$ (or $\sigma_0(n)$) for $n = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$?
- A) $a_1 + a_2 + \dots + a_k$
 - B) $(a_1 + 1)(a_2 + 1) \dots (a_k + 1)$
 - C) $p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$
 - D) $a_1 a_2 \dots a_k$

12. What is the sum of divisors function $\sigma(n)$ (or $\sigma_1(n)$) for $n = p^a$ where p is prime?

- A) p^a
- B) $\frac{p^{a+1} - 1}{p - 1}$
- C) $a p$
- D) $p^{a-1}(p-1)$

13. In complexity theory, an algorithm whose running time is bounded by a polynomial in the input size (number of bits) is classified in complexity class:

- A) EXP
- B) P
- C) NP-hard
- D) Undecidable

14. Is the Euclidean gcd algorithm a polynomial-time algorithm with respect to the input bit length?

- A) No, it is exponential
- B) Yes, it runs in polynomial time ($O(b^2)$ or $O(b^3)$)
- C) No, it runs in linear time relative to value
- D) It depends entirely on whether a and b are even

15. What is $\text{gcd}(F_{n+1}, F_n)$ for any Fibonacci number F_n ?

- A) F_n
- B) 1
- C) F_{n-1}
- D) 2

16. If $a, b, c \in \mathbb{N}$ and $a \mid bc$ with $\text{gcd}(a, b) = 1$, what can be strictly deduced?

- A) $a \mid c$
- B) $b \mid c$
- C) $a \mid b$
- D) $c \mid a$

17. What is the multiplicative inverse of $3 \pmod{11}$?

- A) 2
- B) 4
- C) 7

- D) 8

18. If we want to solve the linear congruence $ax \equiv b \pmod{m}$, a solution exists if and only if:

- A) m divides b
- B) $\text{gcd}(a, m)$ divides b
- C) a divides b
- D) $\text{gcd}(a, b) = 1$

19. In the context of algorithms, what is amortized analysis used for?

- A) Finding prime numbers
- B) Analyzing the average performance of each operation over a worst-case sequence of operations
- C) Calculating memory leaks
- D) Sorting lists in reverse order

20. Which ancient Indian text explicitly details the long division and pulverizer (*Kuttaka*) method for solving linear Diophantine equations and GCDs?

- A) Rig Veda
- B) Aryabhatiya by Aryabhata (and later works by Brahmagupta and Bhaskara II)
- C) Sulba Sutras
- D) Lilavati by Bhaskara II only

Olympiad Level MCQ Answer Key

1. B
2. B
3. C
4. B
5. C
6. A
7. C
8. A
9. B
10. D
11. B
12. B
13. B
14. B
15. B
16. A
17. B

18. B
19. B
20. B

Olympiad Level Subjective Questions

1. Prove Lamé's Theorem: The number of division steps in the Euclidean algorithm for inputs $a > b$ is less than 5 times the number of base-10 digits of b .
2. State and prove Bézout's Identity, and explain how the Extended Euclidean Algorithm computes the coefficients x and y .
3. Prove that consecutive Fibonacci numbers (F_{n+1}, F_n) are the worst-case inputs for the Euclidean algorithm, requiring the maximum number of steps for a given magnitude.
4. Define Euler's totient function $\phi(n)$. Prove that $\sum_{d \mid n} \phi(d) = n$.
5. Let a, b, c be positive integers. Prove that $\frac{\text{lcm}(a, b, c)}{\text{gcd}(a, b) \cdot \text{gcd}(b, c) \cdot \text{gcd}(c, a)}$ is false or state the correct identity relating triple lcm and gcd.
6. Using the Extended Euclidean Algorithm, find integers x and y such that $375x + 825y = \text{gcd}(375, 825)$.
7. Solve the linear congruence $375x \equiv 75 \pmod{825}$, finding all incongruent solutions modulo 825.
8. Prove that if $\text{gcd}(a, m) = 1$, then the linear congruence $ax \equiv b \pmod{m}$ has a unique solution modulo m .
9. Analyze the bit complexity of Aryabhata's division algorithm using standard division ($O(b^2)$ per step where b is bit length), proving total time is $O(b^3)$.
10. Define multiplicative functions in number theory. Prove that both the divisor count function $d(n)$ and the sum of divisors function $\sigma(n)$ are multiplicative.
11. State the Chinese Remainder Theorem (CRT) and outline an algorithmic procedure to solve a system of simultaneous linear congruences.
12. Prove that if p is a prime number and a is an integer not divisible by p , then $a^{p-1} \equiv 1 \pmod{p}$ (Fermat's Little Theorem) using residue multiplier arguments.
13. Discuss the concept of computational complexity classes P versus NP and where primality testing and integer factorization traditionally sit.
14. Explain the *Kuttaka* (pulverizer) method developed by Aryabhata and Brahmagupta for solving linear Diophantine equations $ax + by = c$.
15. Let $a, b \in \mathbb{N}$. Prove that $2^{\text{gcd}(a, b)} - 1 = \text{gcd}(2^a - 1, 2^b - 1)$.
16. Construct an algorithm to find all divisors of n in $O(\sqrt{n})$ time and output them in sorted order.
17. Prove that the set of all common divisors of m and n is identical to the set of divisors of $\text{gcd}(m, n)$.
18. Analyze the average number of division steps in the Euclidean algorithm as established by Heilbronn and Dixon (proportional to $\frac{12}{\pi^2} \ln N + O(1)$).
19. Prove that if $a \mid c$ and $b \mid c$ with $\text{gcd}(a, b) = 1$, then $ab \mid c$.

20. Write a rigorous algorithmic specification (pseudocode/structured steps) for the Extended Euclidean Algorithm, tracking quotients and remainder coefficients.

Olympiad Level Subjective Answers

1. Lamé's Theorem Proof: Each step of the Euclidean algorithm reduces the remainder $r_{i+1} < r_i / \phi$ (where $\phi = \frac{1+\sqrt{5}}{2}$). Since $\phi^5 > 10$, every 5 steps reduce the magnitude by at least a factor of 10. Thus number of steps $n < 5k$ where k is the number of decimal digits of b .
2. Bézout's Identity: For any nonzero integers a and b , there exist integers x, y such that $ax + by = \gcd(a, b)$. Proof by well-ordering principle on the set of positive linear combinations $as + bt$. The Extended Euclidean algorithm works backward through the Euclidean steps, expressing each remainder as a linear combination of a and b .
3. Fibonacci Worst-Case Proof: The Euclidean algorithm on Fibonacci numbers F_{n+1} and F_n takes $n-1$ division steps because $F_{n+1} = 1 \cdot F_n + F_{n-1}$, and remainders are successive Fibonacci numbers. By Lamé's analysis, any numbers requiring n steps must be at least as large as F_{n+2} and F_{n+1} , proving Fibonacci numbers are the minimum magnitude inputs for n steps.
4. $\phi(n)$ is the number of integers in $\{1, 2, \dots, n\}$ coprime to n . Proof of $\sum_{d \mid n} \phi(d) = n$: Partition the fractions $1/n, 2/n, \dots, n/n$ by reducing each fraction to lowest terms a/d where $d \mid n$ and $\gcd(a, d) = 1$. There are $\phi(d)$ such numerators for each divisor d , summing to n .
5. The correct identity for triple lcm is: $\text{lcm}(a, b, c) = \frac{abc \cdot \gcd(a, b, c)}{\gcd(a, b) \cdot \gcd(b, c) \cdot \gcd(c, a)}$ is false; the correct identity is $\text{lcm}(a, b, c) = \frac{abc \cdot \gcd(a, b, c)}{\gcd(a, b) \cdot \gcd(b, c) \cdot \gcd(c, a)}$. Wait, standard relation is $\text{lcm}(a, b, c) = \frac{abc \cdot \gcd(a, b, c)}{\gcd(a, b) \cdot \gcd(b, c) \cdot \gcd(c, a)}$ or prime exponent analysis $\max(a_i, b_i, c_i)$.
6. For 375 and 825: $\gcd(375, 825) = 75$. Euclidean steps: $825 = 2(375) + 75$, $375 = 5(75) + 0$. Working backwards: $75 = 825 - 2(375)$, so $x = -2, y = 1$ ($375(-2) + 825(1) = 75$).
7. $375x \equiv 75 \pmod{825}$. Since $\gcd(375, 825) = 75$, divide entire congruence by 75: $5x \equiv 1 \pmod{11}$. Modulo 11 inverse of 5 is 9 ($5 \cdot 9 = 45 \equiv 1$). Thus $x \equiv 9 \pmod{11}$. Incongruent solutions modulo 825: $9, 9 + 115, 9 + 230, \dots$ (75 solutions total).
8. If $\gcd(a, m) = 1$, by Bézout's identity there exist x, y such that $ax + my = 1$, so $ax \equiv 1 \pmod{m}$. Multiplying by b gives $bx \equiv ab^{-1} \pmod{m}$, yielding a unique solution modulo m .
9. Bit complexity: Each division step on bit lengths b takes $O(b^2)$ bit operations (long division). By Lamé's theorem, there are $O(b)$ division steps. Total time = $O(b) \times O(b^2) = O(b^3)$.
10. A function f is multiplicative if $f(mn) = f(m)f(n)$ whenever $\gcd(m, n) = 1$. Proof for $d(n)$ and $\sigma(n)$ follows from unique prime factorization and multiplicative properties of divisor sums over coprime prime powers.

11. CRT states that if moduli $m_1, m_2, \dots, m_k$ are pairwise coprime, the system $x \equiv a_i \pmod{m_i}$ has a unique solution modulo $M = m_1 m_2 \dots m_k$, constructed using Bézout coefficients and partial products $M_i = M / m_i$.
12. Fermat's Little Theorem: Consider the set $\{a, 2a, 3a, \dots, (p-1)a\}$. When reduced modulo p , these residues are distinct and equal to $1, 2, \dots, p-1$ in some order. Taking their product gives $a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$. Canceling $(p-1)!$ yields $a^{p-1} \equiv 1 \pmod{p}$.
13. P is the class of problems solvable in polynomial time; NP is the class verifiable in polynomial time. Primality testing was proven to be in P (AKS algorithm in 2002); integer factorization is strongly believed to be in NP but not in P (forming the basis of RSA cryptography).
14. The *Kuttaka* (pulverizer) is an ancient Indian algebraic method (Aryabhata, Brahmagupta) using Euclidean division remainders to solve linear Diophantine equations $ax + by = c$, effectively finding integer multipliers.
15. Proof of $2^{\text{gcd}(a, b)} - 1 = \text{gcd}(2^a - 1, 2^b - 1)$ relies on the division algorithm applied to exponents a and b , noting that $2^{qm+r} - 1 = (2^{qm} - 1)2^r + (2^r - 1)$, establishing a Euclidean reduction chain for Mersenne numbers.
16. $O(\sqrt{n})$ divisor algorithm: Loop i from 1 up to $\lfloor \sqrt{n} \rfloor$. If $i \mid n$, add i and n/i to a list. Remove duplicates, sort the list in ascending order, and return.
17. Any common divisor of m and n must divide their greatest common divisor $\text{gcd}(m, n)$ by definition of gcd. Conversely, any divisor of $\text{gcd}(m, n)$ divides both m and n . Thus the two sets of divisors are identical.
18. Heilbronn and Dixon proved that the average number of division steps in the Euclidean algorithm for inputs less than N approaches $\frac{12 \ln 2}{\pi^2} \ln N + \text{Euler-Mascheroni corrections}$, showing logarithmic average growth.
19. If $a \mid c$, then $c = ka$. Since $b \mid c$, $b \mid ka$. Because $\text{gcd}(a, b) = 1$, by Euclid's lemma $b \mid k$, so $k = jb$, meaning $c = jab$, so $ab \mid c$.
20. Extended Euclidean Algorithm Pseudocode: **ExtGCD(a, b)**: if $b = 0$ return $(a, 1, 0)$; else $(d, x_1, y_1) = \text{ExtGCD}(b, a \bmod b)$, return $(d, y_1, x_1 - \lfloor a/b \rfloor y_1)$.