Fibonacci, Geometry and Intro to Combinatorics

Course on Mathematics & Puzzles for Interview Preparation

Arjun Arul • Lesson 3 • Feb 13, 2021

Maths and Puzzles

Lecture 1

Maths and Puzzles

Lecture 2

CONCEPT

Binary Exponentiation

Recurrence Relation

$$a^n = \begin{cases} 1 & n == 0 \\ (a^{n/2})^2 & n\,is\,even\,and\,n > 0 \\ a \cdot (a^{\frac{n-1}{2}})^2 & n\,is\,odd\,and\,n > 0 \end{cases}$$

Recursive Implementation

```
function binpow(a, b) {
    if (b == 0)
        return 1
    let res = binpow(a, b / 2)
    if (b % 2)
        return res * res * a
    else
        return res * res
}
```

Iterative Implementation

```
function binpow(a, b) {
    let res = 1
    while (b > 0) {
        if (b & 1)
            res = res * a
        a = a * a
        b = b / 2
    }
    return res
}
```

QUIZ

Binary Exponentiation

1. Best Time complexity of A^N Exponentiation that I can code is -

A. O(N)
B. O(sqrt(N))
C. O(log(N))
D. O(1)

2. Binary Exponentiation of A^B will fail in which case?

A. When A is Prime number
B. When B is Odd power
C. When GCD(A,B) = 1
D. None of the above

CONCEPT

Modular Arithmetic

Properties of Modular Arithmetic

$$(a + b) \mod n = ((a \mod n) + (b \mod n)) \mod n$$

$$(a \cdot b) \mod n = ((a \mod n) \cdot (b \mod n)) \mod n$$

$$(a - b) \mod n = ((a \mod n) - (b \mod n) + n) \mod n$$

$$(a/b) \mod n \neq ((a \mod n)/(b \mod n)) \mod n$$

QUIZ

Modular Arithmetic

3. The value of A mod B always lies in the set -

A. {0,A-1}
B. {A,B-1}
C. {0,B}
D. {0,B-1}

4. If A belongs to the set {220,330}, Range of (A mod 100) + 10, will be -

A. {0,99}
B. {0,0}
C. {10,109}
D. {10,99}

5. We can apply chaining modulo for the following calculation -

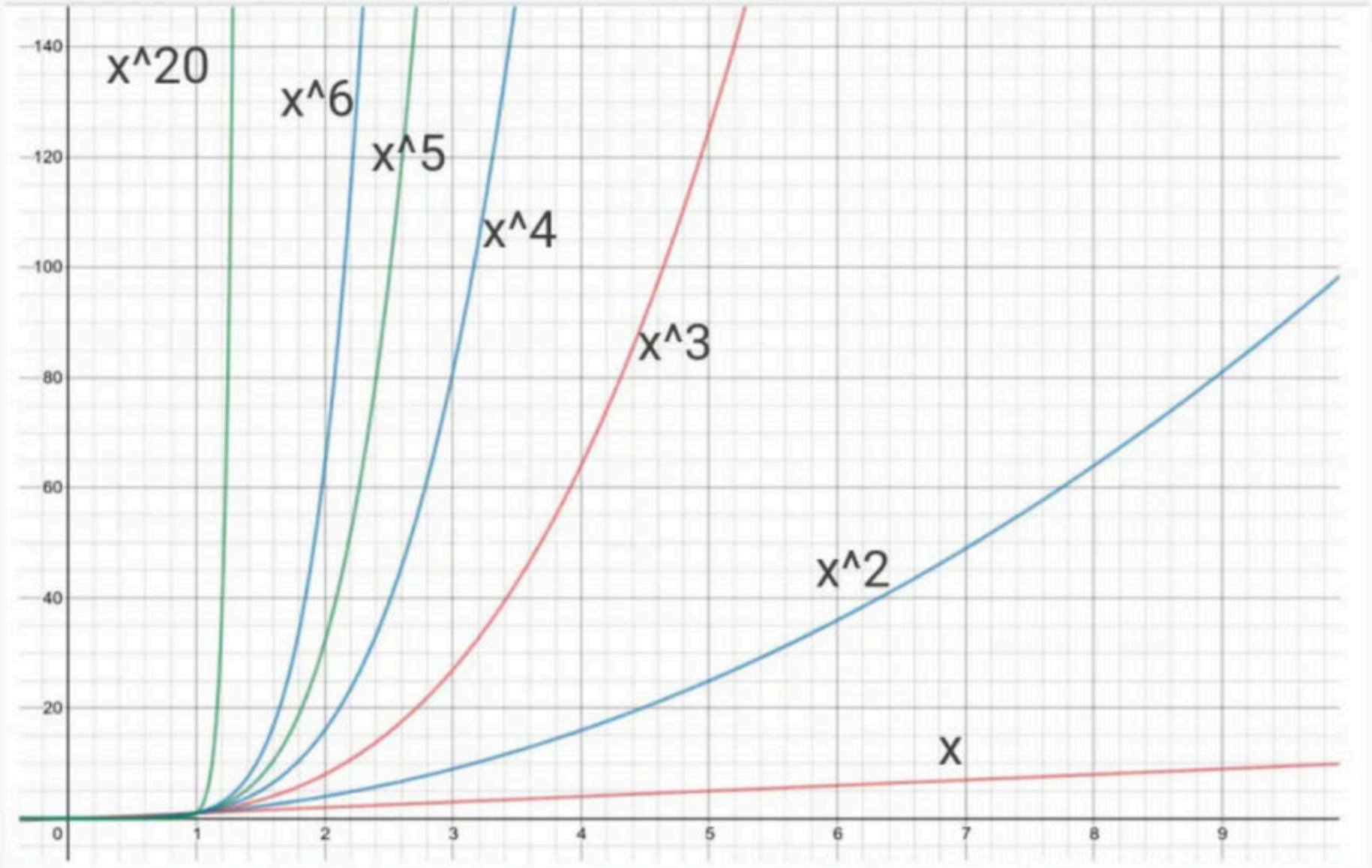
$$(10 - 20 + 24 - 13 - 43) \bmod (17)$$

A. True
B. False

CONCEPT

Modular Exponentiation

Exponential growth comparison



Modular Exponentiation

```
long long binpow(long long a, long long b, long long m) {
    a %= m;
    long long res = 1;
    while (b > 0) {
        if (b & 1)
            res = res * a % m;
        a = a * a % m;
        b = b / 2;
    }
    return res;
}
```

CONCEPT

Fermat’s little theorem

Fermat's Little Theorem

If P is a prime number, $p|(a^p - a)$

Which can be written as $a^p \equiv a \mod p$

Therefore, $a^{p-1} \equiv 1 \mod p$

Fermat's Little Theorem is special case of Euler's Theorem, for prime numbers $a^{\varphi(n)} \equiv 1 \mod n$

QUIZ

Fermat’s little theorem

6. Find the value of x

$$a^{18} \equiv x \mod 7$$

A. 0
B. 1
C. a
D. Can’t determine

7. Find the value of x

$$p^{(p-1)} \equiv x \mod p$$

A. 0
B. 1
C. p
D. Can't determine

CONCEPT

Modulo Multiplicative Inverse and Calculation

Modulo Multiplicative Inverse

If, $b \cdot x \equiv 1(\mod p)$

Then, $(a \div b) \mod p \equiv (a \cdot x) \mod p$

If M is a prime number, then MMI of x is $x^{\varphi(m)-1}$

For a prime 'm', we have $\varphi(m) = m - 1$

Therefore, MMI of X under M is x^{m-2}

MMI Implementation

```
Function mmi(a, m) {
    //m here should be prime
    let b = m-2;
    let res = 1;
    while (b > 0) {
        if (b & 1)
            res = res * a;
        a = a * a;
        b = b / 2;
    }
    return res;
}
```

QUIZ

MMI and Calculation

8. Using the procedure explained, we can find MMI of ‘x’ under modulo ‘m’ in following cases

A. When x and m both are prime
B. When x is composite and m is prime
C. When only x is prime
D. When m is prime

9. Time complexity to find the MMI of ‘x’ under modulo ‘m’ will be

A. O(log(m-2))
B. O(m log(m))
C. O(log(x))
D. O(log(x-2))

CONCEPT

Prime Numbers and Prime Factorisation

Prime Factorisation

$$n = p_1^{a_1} \cdot p_2^{a_2} \cdot p_3^{a_3} \cdots p_k^{a_k}$$

Results from Prime Factorisation of a number

$$\tau(n) = \prod_{i=1}^{k} (a_i + 1)$$

Number of All Factors

$$\sigma(n) = \prod_{i=1}^{k} (1 + p_i + \cdots + p_i^{a_i})$$

Sum of All Factors

$$\mu(n) = n^{\tau(n)/2}$$

Product of All Factors

QUIZ

Prime Numbers and Prime Factorisation

1. I understand what is Prime Factorisation

A. Very Clear
B. Somewhat Clear
C. Please explain again

2. In the given Prime Factorisation, p1 can be same as p2.

$$n = p_1^{a_1} \cdot p_2^{a_2}$$

A. True
B. False

2. In the given Prime Factorisation, p1 can be same as p2.

$$n = p_1^{a_1} \cdot p_2^{a_2}$$

A. True
B. False

Bonus: Can a_1 be same as a_2 ?

CONCEPT

Checking Primality

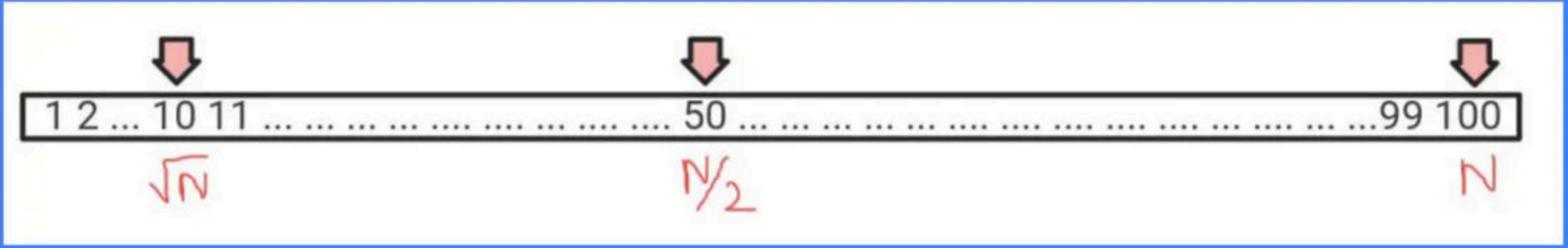
Naive Primality Test

```
prime(n) :
  if n < 2 :
    return false
  loop x=2 upto x=n-1 :
    if x divides n :
      return false
  //return outside loop
  return true
```

Sqrt(N) Primality test

```
prime(n) :
  if n < 2 :
    return false
  loop x=2, x*x<=n, x++ :
    if x divides n:
      return false;
  //outside loop
  return true
```

Sqrt(N) is a big improvement!

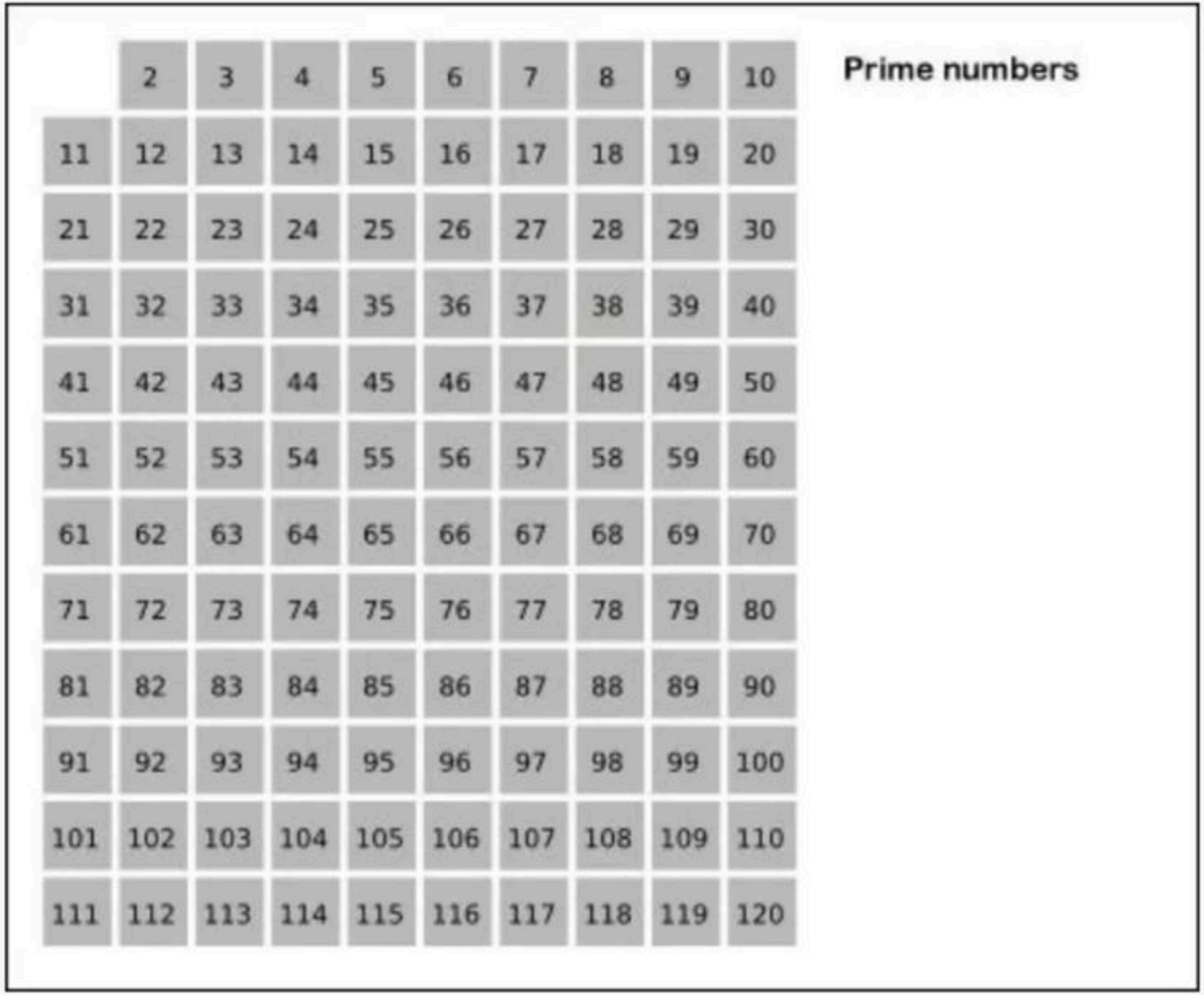


CONCEPT

Sieve of Eratosthenes

Filling the sieve

LINK TO GIF



Sieve of Eratosthenes

```
booleanArray is_prime(n+1, true);
is_prime[0] = is_prime[1] = false;
for (int i = 2; i <= n; i++) {
    if (is_prime[i] == true) {
        for (int j = i; j <= n; j += i)
            is_prime[j] = false;
    }
}
```

Sieve of Eratosthenes - Optimisation

```
booleanArray is_prime(n+1, true);
is_prime[0] = is_prime[1] = false;
for (int i = 2; i * i <= n; i++) {
    if (is_prime[i] == true) {
        for (int j = i * i; j <= n; j += i)
            is_prime[j] = false;
    }
}
```

Sieve of Eratosthenes - Analysis

TIME COMPLEXITY

$$O(n \log \log n)$$

SPACE COMPLEXITY

$$O(n)$$

QUIZ

Checking Primality and Sieve of Eratosthenes

3. Best time complexity that I know, to check if a number is prime is -

A. O(n^2)
B. O(n)
C. $O(\log_2 n)$
D. $O(\sqrt{n})$

4. Best way to generate list of first 1000 primes is

A. Using the Sieve of Eratosthenes
B. Using O(N) Primality check on each number
C. Using O(√N) Primality check on each number
D. Randomly generating a list

CONCEPT

GCD and LCM

Properties of GCD and LCM

$$a * b = \gcd(a, b) * lcm(a, b)$$

$$\gcd(a_1, a_2, \cdots, a_n) = \gcd(a_1, \gcd(a_2, a_3, \cdots, a_n))$$

$$lcm(a_1, a_2, \cdots, a_n) = lcm(a_1, lcm(a_2, a_3, \cdots, a_n))$$

$$\gcd(m \cdot a, m \cdot b) = m \cdot \gcd(a, b)$$

$$1 \leq \gcd(a, b) \leq min(a, b)$$

$$max(a, b) \leq (a, b) \leq a \cdot b$$

QUIZ

GCD and LCM

5. What is the value of y in the following equation?

$$\gcd(a \cdot x, b \cdot x) = y \cdot \gcd(a, b)$$

A. x
B. x*x
C. a*b*x
D. x/(a*b)

6. What is the output, if GCD(A,B) = 1. (A and B both are greater than 1)

```
boolean array[B+1]
setAll(array,false)
loop i=A, i<=B, i+=A :
    array[i] = true
print(array[B])
```

A. 0
B. 1
C. Can’t Determine

CONCEPT

Euclid’s Algorithms

$\rightarrow gcd(a, b) = gcd(b, a \% b)$

$a \geq b$

$a = 4b + (a \% b)$

$a \% b = (a - 4b)$

Any common divisor of a & b is also a divisor of $a \cdot b$. $\rightarrow \gcd(a, b) \leq \gcd(b, a \cdot b)$

Any common divisor of $(a \cdot b)$ & b is also a divisor of a.

$\rightarrow \gcd(b, a \cdot b) \leq \gcd(a, b)$

$$gcd(a, b) = gcd(b, a \% b)$$

$$gcd(a, b) = gcd(b, a \% b)$$

$gcd(a, 0) = a \rightarrow$ Base Case.

$a = 21 \rightarrow 1, 3, 7, 21$

$b = 14 \rightarrow 1, 2, 7, 14$

$GCD(a, b) = \{1, 7\}$

$a \% b = 21 \% 14 = 7 \rightarrow \{1, 7\}$

$GCD(14, 7) = \{1, 7\}$

Recurrence Relation

$$\gcd(a + m \cdot b, b) = \gcd(a, b)$$

$$\gcd(a, b) = \begin{cases} a & b = 0 \\ \gcd(b, a \mod b) & b \neq 0 \end{cases}$$

Visualizing Euclid's Algorithm

1071 x 462

147 x 147

LINK TO GIF

462 x 462

21 x 21

Finding GCD and LCM by Euclid's Algorithm

```
gcd(a, b) :
    if b == 0 : return a
    else return gcd(b, a%b)
```

$a \times b = g \times l$

$l = \frac{a b}{g}$

```
lcm(a, b) :
   return a / gcd(a, b) * b
```

$$gcd(a, b) \longrightarrow gcd(b, a \bmod b)$$

$$\downarrow$$

$$gcd(a, b) = gcd(a \bmod b, b \bmod (a \bmod b))$$

$$a \bmod b \leq \frac{a}{2}$$

$$\leq 2 \log a$$

$a \mathbin{/\!.} b$

1. Suppose $b \leq \frac{a}{2}$

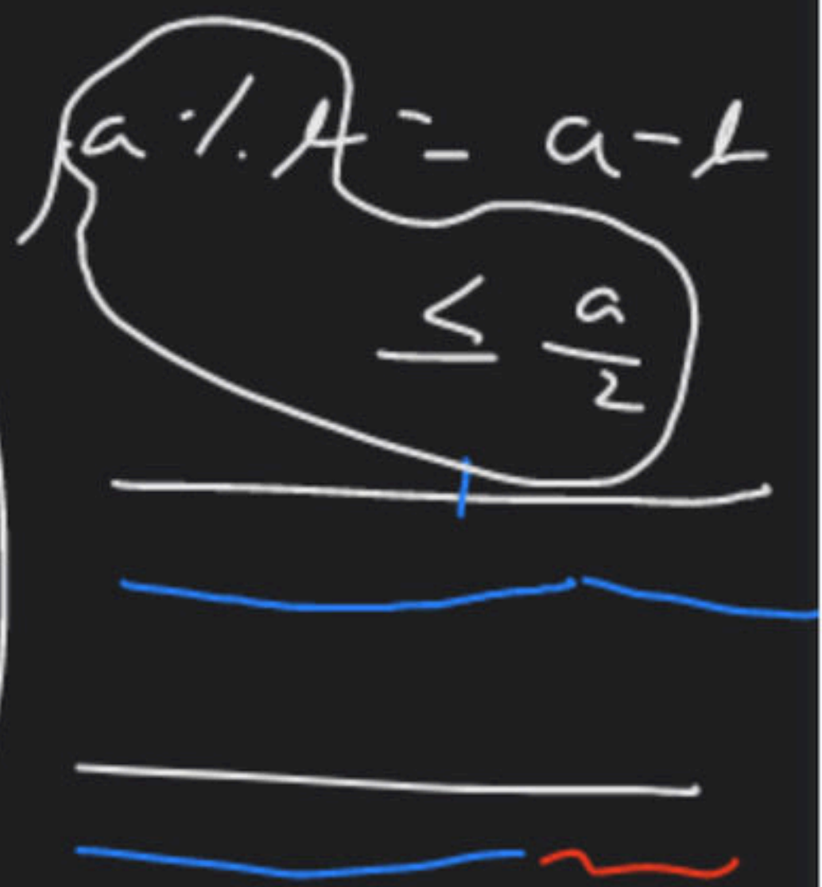
$a \mathbin{/\!.} b_0 < b \leq \frac{a}{2}$

$a \mathbin{/\!.} b \leq \frac{a}{2}$

2. $b > \frac{a}{2}$

$a \mathbin{/\!.} b = a - b$

$\leq \frac{a}{2}$



$a \% b \leq \frac{a}{2}$

$O(\log M)$

$O(\log b)$

Bezout's Theorem

If $\gcd(a, b) = g$

$0 < (xa + yb) < g$, x, y are integers.

$x = b$ $\quad ba - ab = 0$

$y = -a$

$$a\underline{i} + b\underline{j} = \underline{s}$$

$$2(2a\underline{i}) + 1(2b\underline{j}) = 2\underline{s} \quad 2a\underline{s}$$

$gcd(a, b) = 1 \rightarrow$ co-prime

$ax + by = 1$

21, 16

3×7

2^4

3, 7

$7 \times 1 + 3 \times (-2) = 1$

Extended Euclid's Algorithm

```
int gcd(int a, int b, int& x, int& y)
{
    if (b == 0) {
        x = 1;
        y = 0;
        return a;
    }
    int x1, y1;
    int d = gcd(b, a % b, x1, y1);
    x = y1;
    y = x1 - y1 * (a / b);
    return d;
}
```

$$a \cdot x + b \cdot y = \gcd(a, b)$$

$x_1, b + y_1, (a \cdot / \cdot b) = gcd(b, a \cdot / \cdot b)$

$= g$

$\boxed{a x + b y = g}$

$\boxed{a \cdot / \cdot b = a - \left\lfloor \frac{a}{b} \right\rfloor \times b}$

$a = \left\lfloor \frac{a}{b} \right\rfloor \times b + (a \cdot / \cdot b)$

$$x_1 r + y_1 \left[a - \left\lfloor \frac{a}{r} \right\rfloor r \right] = g$$

$$x_1 r + y_1 a - y_1 \left\lfloor \frac{a}{r} \right\rfloor r = g$$

$$y_1 a + r \left(x_1 - y_1 \left\lfloor \frac{a}{r} \right\rfloor \right) = g$$

$$x = y_1$$

$$y = x_1 - y_1 \left\lfloor \frac{a}{x} \right\rfloor$$

$$xa + yb = g$$

$x_1 b + y_1 (a \% b) = g$

$x a + y b = g$ Find x & y

QUIZ

Euclid’s Algorithms

7. Using Euclid's Algorithm we can check if two numbers are Coprime.

A. True
B. False

8. Time complexity for finding the GCD of an array of integers is

$gcd(a,b,c) = gcd(a, gcd(b,c))$

A. $O(N^2)$
B. $O(N \log_2 M)$, where M is maximum value of array
C. $O(N \log_2 N)$
D. $O(N)$

Maths and Puzzles

Lecture 2

CONCEPT

Binary Exponentiation

a, n

a^n

$O(n)$

$5, 2 \rightarrow 25$

$7, 3 \rightarrow 343$

$a^{10} = a^{5} \times a^{5}$

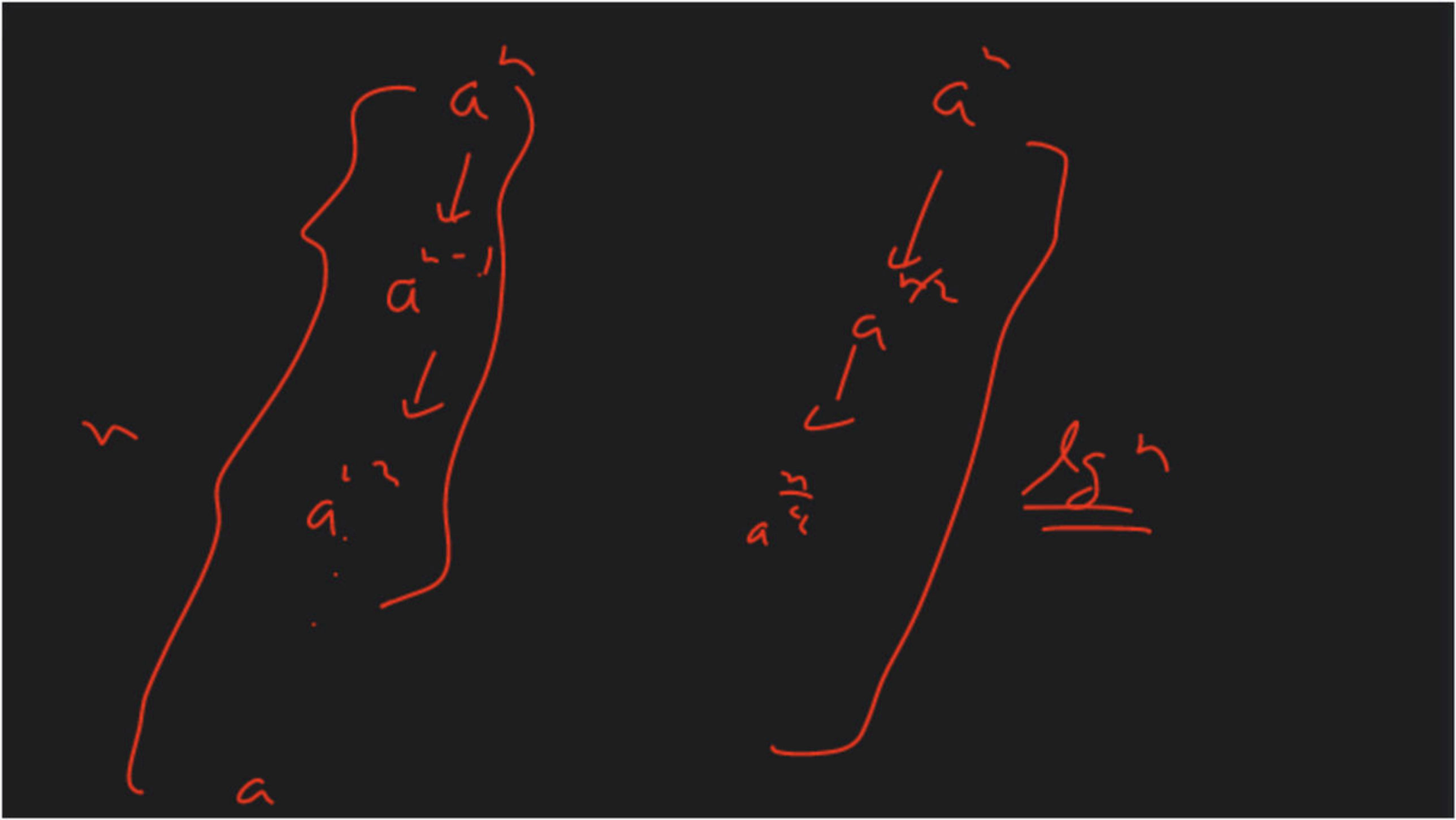
$a^{n} = a^{\frac{n}{2}} \times a^{\frac{n}{2}}$

$a^{1} \times a^{c} = a^{1+c}$

$a^{11} = a^{5} \times a \times a^{5}$

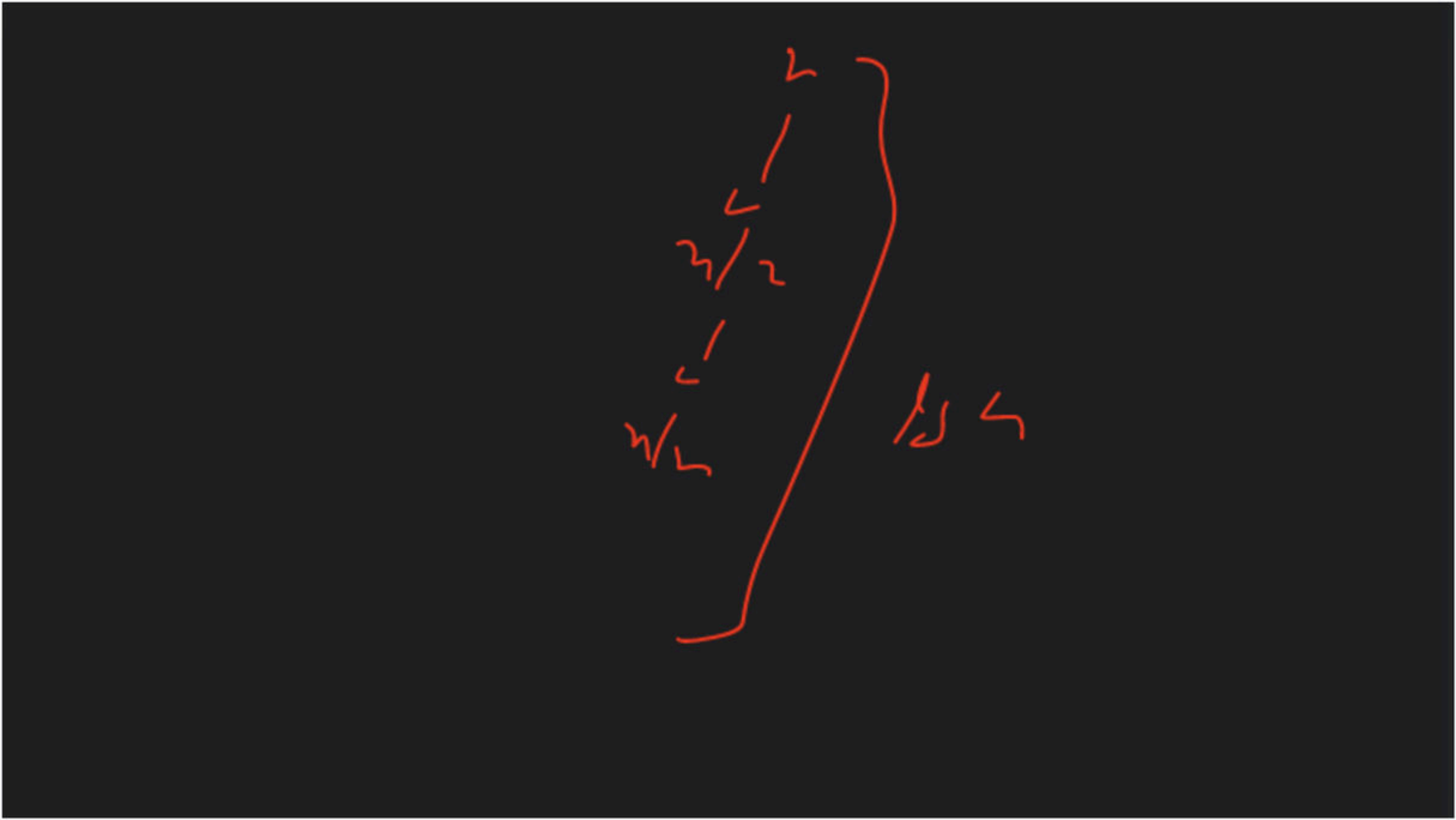
Recurrence Relation

$$a^n = \begin{cases} 1 & n == 0 \\ (a^{n/2})^2 & n\,is\,even\,and\,n > 0 \\ a \cdot (a^{\frac{n-1}{2}})^2 & n\,is\,odd\,and\,n > 0 \end{cases}$$



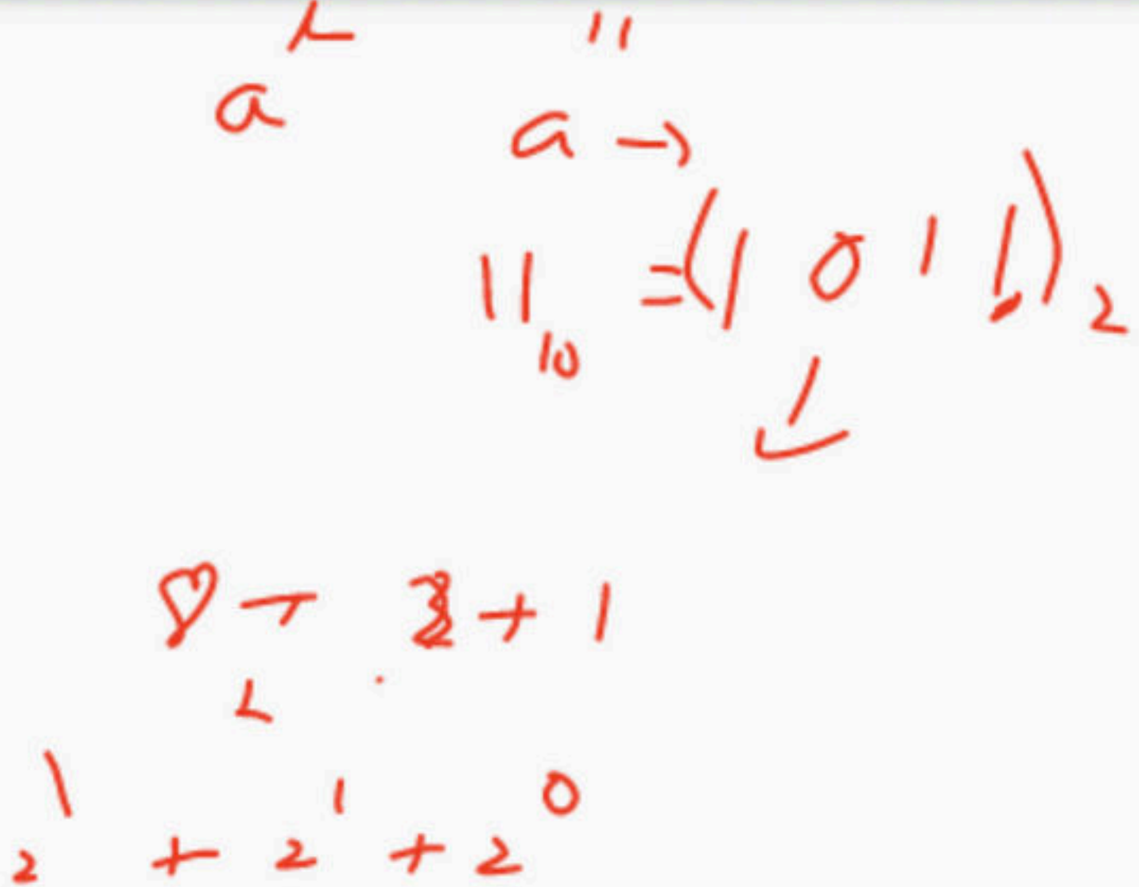
Recursive Implementation

```
function binpow(a, b) {
    if (b == 0)
        return 1
    let res = binpow(a, b / 2)
    if (b % 2)
        return res * res * a
    else
        return res * res
}
```



Iterative Implementation

```
function binpow(a, b) {
    let res = 1
    while (b > 0) {
        if (b & 1)
            res = res * a
        a = a * a
        b = b / 2
    }
    return res
}
```



$$11 = 2^3 + 2^1 + 2^0$$

$$a^{11} = a^{2^3 + 2^1 + 2^0}$$

$$= a^{2^3} \times a^{2^1} \times a^{2^0}$$

QUIZ

Binary Exponentiation

1. Best Time complexity of A^N Exponentiation that I can code is -

A. O(N)
B. O(sqrt(N))
C. O(log(N))
D. O(1)

2. Binary Exponentiation of A^B will fail in which case?

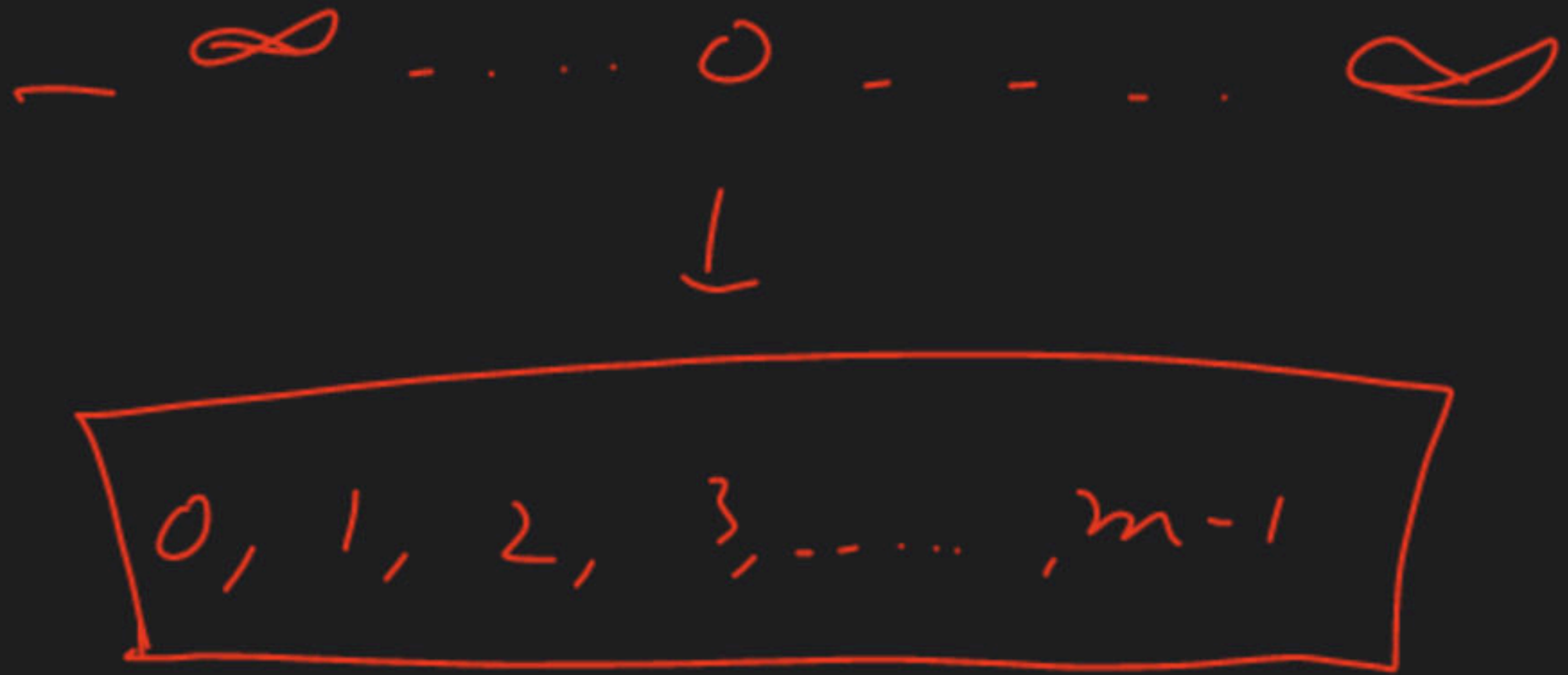
A. When A is Prime number
B. When B is Odd power
C. When GCD(A,B) = 1
D. None of the above

$15 \% 12 = 3$

$15 - 12 = 3$

CONCEPT

Modular Arithmetic


$-\infty \ldots\ldots 0 \ldots\ldots \infty$
$\downarrow$
$0, 1, 2, 3, \ldots\ldots, m-1$
$\bmod m$

Properties of Modular Arithmetic

$$(a + b) \mod n = ((a \mod n) + (b \mod n)) \mod n$$

$$(a \cdot b) \mod n = ((a \mod n) \cdot (b \mod n)) \mod n$$

$$(a - b) \mod n = ((a \mod n) - (b \mod n) + n) \mod n$$

$$(a/b) \mod n \neq ((a \mod n)/(b \mod n)) \mod n$$

$24 + 35$

$(20 + 30) \bmod 9 = (20 \bmod 9) + (50 \bmod 9)$

$1 \qquad\qquad 2 \qquad\qquad 3$

$50 \bmod 9$

1

5

$51 \to 5$

$= (5) \bmod 5$

$= 5$

$= 6 + 8$

$= 14 = 5$

QUIZ

Modular Arithmetic

3. The value of A mod B always lies in the set -

15 mod 9

A. {0,A-1}
B. {A,B-1} ×
C. {0,B}
D. {0,B-1} ✓

[0, 1)

[0, B-1)

4. If A belongs to the set {220,330}. Range of (A mod 100) + 10, will be -

(0, ... 99)

(10, --- 109)

A. {0,99}
B. {0,0}
C. {10,109} ✓
D. {10,99}

5. We can apply chaining modulo for the following calculation -

$$(10 - 20 + 24 - 13 - 43) \bmod (17)$$

A. True
B. False

Congruence.

$$10 \equiv 1 \cdot 9 \pmod 9$$

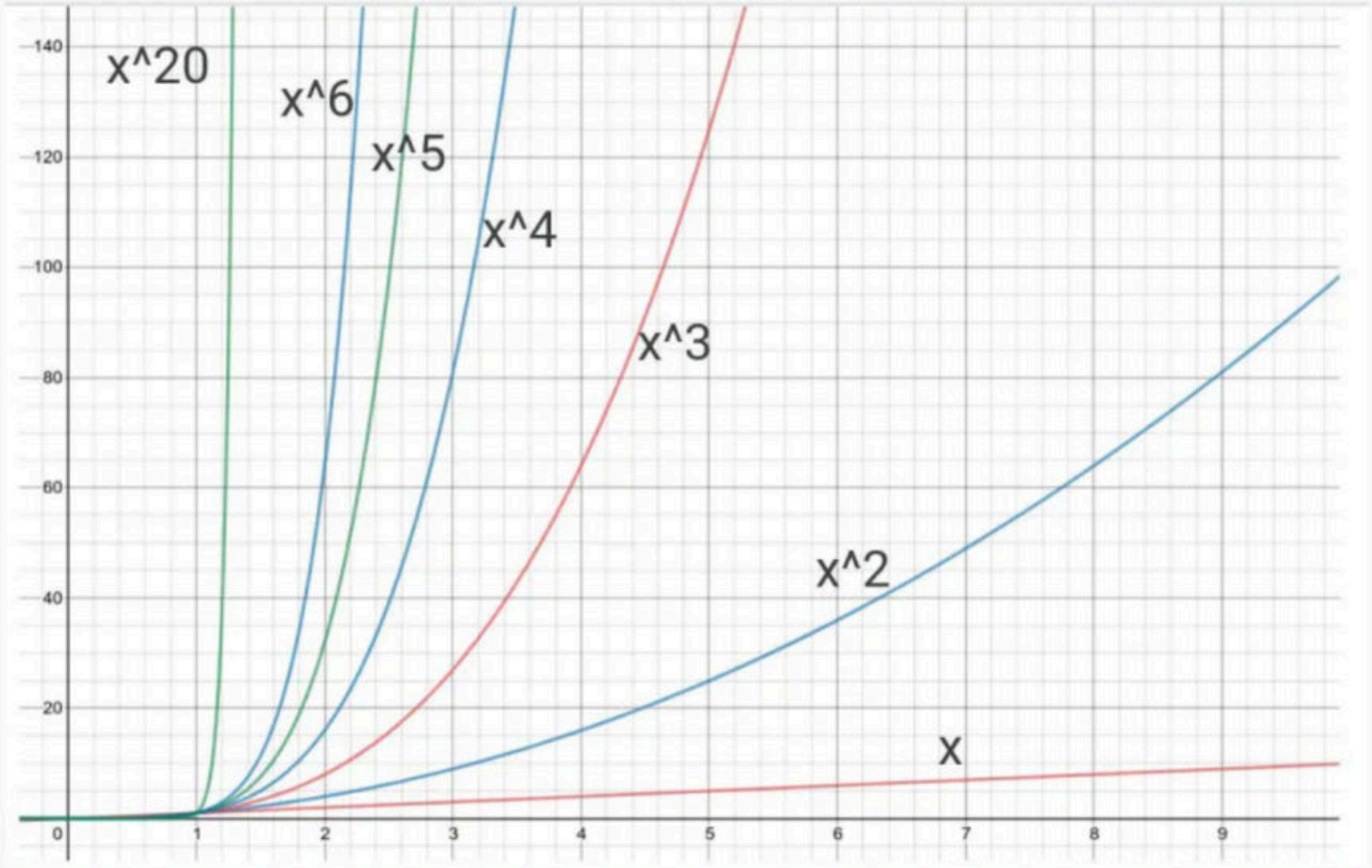
$$1 \equiv 1$$

$$2! = 2 \cdot 1 = 4$$

CONCEPT

Modular Exponentiation

Exponential growth comparison



Modular Exponentiation

```
long long binpow(long long a, long long b, long long m) {
    a %= m;
    long long res = 1;
    while (b > 0) {
        if (b & 1)
            res = res * a % m;
        a = a * a % m;
        b = b / 2;
    }
    return res;
}
```

CONCEPT

Fermat’s little theorem

Fermat's Little Theorem

If P is a prime number, $p|(a^p - a)$

Which can be written as $a^p \equiv a \mod p$

Therefore, $a^{p-1} \equiv 1 \mod p$

Fermat's Little Theorem is special case of Euler's Theorem, for prime numbers $a^{\varphi(n)} \equiv 1 \mod n$

QUIZ

Fermat’s little theorem

6. Find the value of x

$$a^{18} \equiv x \mod 7$$

A. 0
B. 1
C. a
D. Can’t determine

7. Find the value of x

$$p^{(p-1)} \equiv x \mod p$$

A. 0
B. 1
C. p
D. Can't determine

CONCEPT

Modulo Multiplicative Inverse and Calculation

Modulo Multiplicative Inverse

If, $b \cdot x \equiv 1(\mod p)$

Then, $(a \div b) \mod p \equiv (a \cdot x) \mod p$

If M is a prime number, then MMI of x is $x^{\varphi(m)-1}$

For a prime 'm', we have $\varphi(m) = m - 1$

Therefore, MMI of X under M is x^{m-2}

MMI Implementation

```
Function mmi(a, m) {
    //m here should be prime
    let b = m-2;
    let res = 1;
    while (b > 0) {
        if (b & 1)
            res = res * a;
        a = a * a;
        b = b / 2;
    }
    return res;
}
```

QUIZ

MMI and Calculation

8. Using the procedure explained, we can find MMI of ‘x’ under modulo ‘m’ in following cases

A. When x and m both are prime
B. When x is composite and m is prime
C. When only x is prime
D. When m is prime

9. Time complexity to find the MMI of ‘x’ under modulo ‘m’ will be

A. O(log(m-2))
B. O(m log(m))
C. O(log(x))
D. O(log(x-2))