Competitive Programming

Math

B.Bhuvaneswaran, AP (SG) / CSE

9791519152

bhuvaneswaran@rajalakshmi.edu.in

RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Math

- To crack programming interviews and coding challenges, it is not necessary for you to learn the in-depth of all math concepts.
- However, you do need to learn some topics which are useful to solve many coding questions.
- The topics which are most commonly asked are based on:
  - Number Systems
  - Basic Operations
  - Quotient Remainder Theorem

Number Systems

- The base of any number system is determined by the number of digits in the system.
- In our day to day life, we use the decimal system.
- In the decimal system, we have only 10 digits: 0 ,1, 2, 3, 4, 5, 6, 7, 8, 9.
- All the numbers are a combination of these digits.

Number Systems

Number System	Base	Digits
Binary	2	0 , 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Hexa Decimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Number Systems

- There are 2 main concepts to remember when dealing with different number systems:
  - When does a number tick over to the next number?
  - How to read a given number?

Ticking over

- In any number system, we *tick over* when the current position is at it’s maximum possible value in terms of digit.

Example

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9 (we have reached maximum value)
- 10 (we ticked over)

Ticking over

Decimal	Binary	Octal	Hexa
0	0	0	0
1	1	1	1
2	10 (ticked over)	2	2
3	11	3	3
4	100 (ticked over)	4	4
5	101	5	5
6	110 (ticked over)	6	6
7	111	7	7
8	1000 (ticked over)	10 (ticked over)	8
9	1001	11	9
10 (ticked over)	1010 (ticked over)	12	A
11	1011	13	B
12	1100 (ticked over)	14	C
13	1101	15	D
14	1110 (ticked over)	16	E
15	1111	17	F
16	10000 (ticked over)	20 (ticked over)	10 (ticked over)
17	10001	21	11
18	10010 (ticked over)	22	12
19	10011	23	13
20 (ticked over)	10100 (ticked over)	24	14

How to read a given number?

- Given a number in any base, we would like to understand what is it’s value in decimal.
- What we have to understand is that, in each number system, the digit represents a factor of the power of the base.
- Don’t worry! It’s not difficult.

Example

- 10 (in binary). What this means is 1*(2^1) + 0*(2^0) = 2 (in decimal)
- Similarly, 1010 (in binary) means 1*(2^3) + 0*(2^2) + 1*(2^1) + 0*(2^0) = 10 (in decimal)
- 14 (in octal) means 1*(8^1) + 4*(8^0) = 12 (in decimal).

Basic Operations

- The basic operations in Math are addition (+), subtraction (-), multiplication (*) and division (/).
- We are sure you have done ample questions involving these.
- The other important operations to remember are the shift operators.
- In computers, all integers are stored in the binary form. The left-shit and right-shift operators shift the bits in this binary form to the left or right respectively.

Example

- 4 >> 1
- means
- 100 >> 1
- = 010
- = 2
- All the bits in 4 are moved to the right.
- The right-most bit is discarded.

Example

- 9 >> 2
- means
- 1001 >> 2
- = 10
- = 2
- All the bits in 9 are moved to the right by 2 places.
- The right-most bit is discarded.
- Similarly, with left-shift operators.

Quotient Remainder Theorem

- Given any integer A, and a positive integer B, there exist unique integers Q and R such that
- $A = B * Q + R$ where $0 \leq R < B$
- We can see that this comes directly from long division. When we divide A by B in long division, Q is the quotient and R is the remainder.
- If we can write a number in this form then A mod B = R

Examples

- A = 7, B = 2
- 7 = 2 * 3 + 1
- 7 mod 2 = 1

Examples

- A = 8, B = 4
- 8 = 4 * 2 + 0
- 8 mod 4 = 0

Examples

- A = 13, B = 5
- $13 = 5 * 2 + 3$
- $13 \bmod 5 = 3$

Examples

- A = -16, B = 26
- -16 = 26 * -1 + 10
- -16 mod 26 = 10

Modular Arithmetic

- Compute your answer modulo 10**9 + 7
- Example
  - 5 modulo 3 → 2
  - 5 % 3 → 2
- Main reason
  - To avoid overflows

Arithmetic Under a Modulo

- x + y mod 5
- (10 + 6) mod 5
- 16 mod 5 → 1
- What is (x + y) if overflowing?

Arithmetic Under a Modulo

- Addition
  - (x + y) mod p = (x mod p + y mod p) mod p
- Subtraction
  - (x - y) mod p = (x mod p - y mod p) mod p
- Multiplication
  - (x * y) mod p = (x mod p * y mod p) mod p
- Note:
  - p must be a prime!

n^2

n	Sum of First 'n' Odd Numbers	?
1	1	1
2	1 + 3	4
3	1 + 3 + 5	9
4	1 + 3 + 5 + 7	16
5	1 + 3 + 5 + 7 + 9	25
6	1 + 3 + 5 + 7 + 9 + 11	36
7	1 + 3 + 5 + 7 + 9 + 11 + 13	49
8	1 + 3 + 5 + 7 + 9 + 11 + 13 + 15	64
9	1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17	81
10	1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19	100

$n * (n + 1)$

n	Sum of First 'n' Even Numbers	?
1	2	2
2	2 + 4	6
3	2 + 4 + 6	12
4	2 + 4 + 6 + 8	20
5	2 + 4 + 6 + 8 + 10	30
6	2 + 4 + 6 + 8 + 10 + 12	42
7	2 + 4 + 6 + 8 + 10 + 12 + 14	56
8	2 + 4 + 6 + 8 + 10 + 12 + 14 + 16	72
9	2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18	90
10	2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20	110

Sum of First 10 Natural Numbers

1	3	5	7	9
2	4	6	8	10

Solution

- n * n
- 5 * 5
- n * (n + 1)
- 5 * 6
- 5 * 11 = 55
- n = 10 → 5 * 11?
- n / 2 * (n + 1)

Sum of First 11 Natural Numbers

1	3	5	7	9	11

2	4	6	8	10

Solution

- n * n
- 6 * 6
- n * (n + 1)
- 5 * 6
- 6 * 11 = 66
- n = 11 → 6 * 11?
- (n + 1) / 2 * n

Conclusion

- $(n + 1) / 2 * n = n / 2 * (n + 1) = n * (n + 1) / 2$

Example

- Given the sum of two consecutive numbers, find the two numbers?

Input	Output	
11	5	6
77	38	39
101	50	51
27	?	?
47	?	?

Solution

- $x, x + 1$
- $x + x + 1 = 27$
- $2x + 1 = 27$
- $2x = 27 - 1$
- $2x = 26$
- $x = 26 / 2 = 13$
- $x + 1 = 14$
- x = input / 2

Example

- Given the sum of three consecutive numbers, find the three numbers?

Input	Output		
18	5	6	7
33	10	11	12
69	22	23	24
27	?	?	?
81	?	?	?

Solution

- $x - 1, x, x + 1$
- $x - 1 + x + x + 1 = 81$
- $3x = 81$
- $x = 81 / 3 = 27$
- $x - 1 = 26$ and $x + 1 = 28$
- x = input / 3

Magic Box

- 1 → 2
- 2 → 1
- What is the magic box doing?
- Without any checking?

Solution

- 3 - input → (sum - input)
- 2 / input → (product / input)
- $2^{2 - \text{input}}$

What is $^{n}P_{r}$?

- $^{n}P_{r}$ means the number of permutations possible when taking r elements at a time from n elements.
- For example, given 3 alphabets A B C, in how many ways can we choose 2 at a time?
- The ways are:
  - AB
  - BA
  - AC
  - CA
  - BC
  - CB

What is ${}^{n}P_{r}$?

- Thus ${}^{3}P_{2}$ is 6
- Notice that, in ${}^{n}P_{r}$, the order of arrangement matters! Here AB and BA are not the same and are hence counted differently.
- The formula for ${}^{n}P_{r}$ is $((n!) / (n - r)!)$
- ${}^{n}P_{r}$ is also referred as $P(n, r)$

What is ^nC_r?

- ^nC_r means the number of combinations possible when taking r elements at a time, but this time the order of arrangement does not matter.
- Thus, choosing any 2 alphabets from A B C, such that the order doesn't matter, we get:
  - AB
  - BC
  - AC

What is ^nC_r?

- Thus $^3C_2$ is 3.
- The formula for ^nC_r is ((n!) / ((r!) (n-r)!)
- ^nP_r is also referred as C(n, r)
- C(n, r) = P(n, r) / r!

Calculating ${}^nC_r$ and ${}^nP_r$

- We are interested to know how to calculate the values of C(n,r) and P(n, r) using code.
- There are many ways to do this. The nature of both C and P is that they can be computed recursively.
- $P(n, k) = P(n - 1, k) + k * P(n - 1, k - 1)$
- $C(n, k) = C(n - 1, k - 1) + C(n - 1, k)$

Calculating $^{n}C_{r}$ and $^{n}P_{r}$

- Instead, let us look at a very common use case: finding C(n,r) where both n and r are large and can lead to overflows.
- In such cases, we opt to always take C(n,r) % p where p is a very large prime number.
- Using the property of modulo, we get:
- C(n, k) % p = C(n - 1, k - 1) % p + C(n - 1, k) % p
- And for P(n,r), we get:
- P(n, k) % p = P(n - 1, k) % p + ((k % p) * (P(n - 1, k - 1) % p) % p)

Queries?

Thank You…!