Agenda :

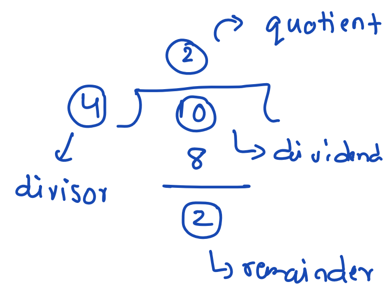
i) Modular arithmetic basics

ii) what is subsequence and subset, questions on them.

Basics of Mod %

$a \% m$ = remainder when a is divided by m

$10 \% 4 = 2$



dividend = divisor * quotient + rem

rem = $\underbrace{\text{dividend - divisor * quotient}}$

↳ greatest multiple of divisor <= dividend

$10 \% 4 = 10 - (8) = 2$

$47 \% 6 = 47 - (\overset{42}{\text{greatest multiple of } 6 <= 47}) = 5$

$-47 \% 6 = -47 - (\overset{-48}{\text{greatest multiple of } 6 <= -47})$

$= -47 - (-48) = 1$

$-55 \% 7 = -55 - (-56) = 1$

$-50 \% 6 = -50 - (-54) = 4$

Issue	Python		Java / C++
$-47 \% 6$	1	$\leftarrow$	$-5 + 6$
$-55 \% 7$	1	$\leftarrow$	$-6 + 7$
$-50 \% 6$	4	$\leftarrow$	$-2 + 6$

```
int rem = a % m;
if (rem < 0) {
        rem = rem + divisor;
}
```

$ans \% (10^9 + 7)$

↳ to manage large ans in case of overflow

$1 \le n \le 10^6$

find factorial of n ?

$$\left.\begin{matrix} -\infty \\ \\ \\ \infty \end{matrix}\right\} \% 8 \Rightarrow 0 \text{ to } 7$$

$$\left.\begin{matrix} -\infty \\ \\ \\ \infty \end{matrix}\right\} \% p \Rightarrow 0 \text{ to } p-1$$

Properties of %

1) $(a+b) \% m = (a \% m + b \% m) \% m$

$a = 6$
$b = 15$
$m = 4$

$21 \% 4$

$= 1$

$(6 \% 4 + 15 \% 4) \% 4$

$(2+3) \% 4$

$= 1$

2) $(a * b) \% m = (a \% m * b \% m) \% m$

$90 \% 4$

$= 2$

$(6 \% 4 * 15 \% 4) \% 4$

$a = 6$
$b = 15$
$m = 4$

$(2 * 3) \% 4$

$6 \% 4 = 2$

3) $(a-b) \% m = (a \% m - b \% m + m) \% m$

Q.1 Given a, n, p. Find $a^n \% p$?

$a = 3$ $\quad p = 5$ $\quad ans = 3^4 \% 5$

$n = 4$ $\quad = 81 \% 5 = 1$

$1 <= a <= 10^9$

$1 <= n <= 10^5$

$1 <= p <= 10^9$

```
int solve (int a, int n, int p) {
    long ans = 1;
    for (int i = 1; i <= n; i++) {
        ans = (ans * a) % p;
    }              ↓     ↓
                 10^9  10^9
    return (int)(ans % p);
}
```

If doubtful to apply % p or not ⇒ do it.

Subsequence : By removing 0 or more elements from Array.

	0	1	2	3	4	5	
A =	[3	2	1	-4	5	9]	
	✗	✗	✓	✗	✓	✓	{1 5 9}
	✗	✗	✓	✓	✗	✓	{1 -4 9}
	✗	✗	✗	✗	✗	✗	{ }
	✗	✓	✗	✗	✗	✗	{2}

1) continuity does not matter
2) order of indexing matter

A => 3 1 2 $\xrightarrow{\text{Sort}}$ A => 1 2 3

A => 3 1 2	A => 1 2 3	
{ }	{ }	✓
{3}	{3}	✓
{1}	{1}	✓
{2}	{2}	✓
{3 1}	{1 3}	✗
{3 2}	{2 3}	✗
{1 2}	{1 2}	✓
{3 1 2}	{1 2 3}	✗

	0	1	2	3	4	5
A =	[3	2	1	-4	5	9]

valid subseq

3 2 1 -4 ✓

2 1 5 ✓

3 2 -4 9 ✓

3 1 9 5 ✗ (order is incorrect)

Subset : Exactly same as subseq but order does not matter.

A => 3 1 2 $\xrightarrow{\text{Sort}}$ A => 1 2 3

{ }	{ } ✓
{3}	{3} ✓
{1}	{1} ✓
{2}	{2} ✓
{3 1}	{1 3} ✓
{3 2}	{2 3} ✓
{1 2}	{1 2} ✓
{3 1 2}	{1 2 3} ✓

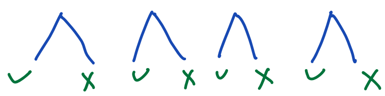
$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$
$$A = \begin{bmatrix} 3 & 2 & 1 & -4 & 5 & 9 \end{bmatrix}$$

3 2 1 5 — subsequence , subset

1 5 2 — subset

count of subsequence

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array}$$
$$A = \begin{bmatrix} 3 & 1 & 2 & -4 \end{bmatrix}$$



$2 * 2 * 2 * 2 = 16$

total subsequence = 2^n

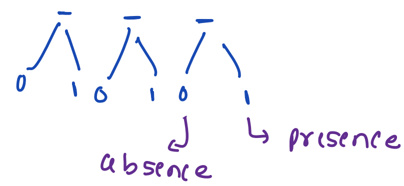
total subset = 2^n
(when array elements are distinct)

Q.1 Given an array (distinct elements) and k. Find if there is any subset with sum = k.

A = [-1 7 5 2 6]   k = 7
(indices: 0 1 2 3 4)

{7} , {5 2} , {-1 2 6}   ans : true

A ⇒ 3 1 4
(indices: 0 1 2)



	2	1	0		
0 →	0	0	0	→	{ }
1 →	0	0	1	→	{3}
2 →	0	1	0	→	{1}
3 →	0	1	1	→	{3 1}
4 →	1	0	0	→	{4}
5 →	1	0	1	→	{3 4}
6 →	1	1	0	→	{1 4}
7 →	1	1	1	→	{3 1 4}

```
boolean solve (int[] A, int k) {
    int n = A.length;
    int tcs = Math.pow(2, n);

    for (int x = 0; x < tcs; x++) {
        // check bits of x from 0 to n-1
         and create your subset sum.
        int sum = 0;

        for (int i = 0; i < n; i++) {
            if( checkbit(x, i) == true) {
                sum += A[i];
            }
        }

        if (sum == k) {
            return true;
        }
    }
    return false;
}
```

k = 5

A = [3 1 -2 4]

n = 4

tcs = 16

x	i	sum
0 (0000)	0-3	0
1 (0001)	0-3	3
2 (0010)	0-3	1
⋮		
9 (1001)	0-3	7
10 (1010)	0-3	5 return true
⋮		
13 (1101)	0-3	5

TC: $O(2^n \times n)$

SC: $O(1)$

Q. 2 Given an array, find sum of max of every subsequence.

A = [3 2 4]

{ } → x

{3} → 3

{2} → 2

{4} → 4

{3 2} → 3

{3 4} → 4

{2 4} → 4

{3 2 4} → 4

24

1) idea1 : using last question

2) idea2 : contribution technique

$3 \times 2 + 2 \times 1 + 4 \times 4$

$= 6 + 2 + 16 = 24$

A[i] is the max of how many subsets.

A = [2 3 4]

2 is max → {2}

3 is max → {2 3} {3}

4 is max → {4} {2 4} {3 4} {2 3 4}

$A = \{1 \quad 2 \quad 3 \quad 7\}$

$\quad\quad\; 0 \quad 1 \quad 2 \quad 3$

1 as max

$\{1\}$

↓

2^0

2 as max

$\{2\}$

$\{1 \; 2\}$

↓

2^1

3 is max

$\{3\}$

$\{1 \; 3\}$

$\{2 \; 3\}$

$\{1 \; 2 \; 3\}$

↓

2^2

7 is max

$\{7\}$

$\{1 \; 7\}$

$\{2 \; 7\}$

$\{1 \; 2 \; 7\}$

$\{3 \; 7\}$

$\{1 \; 3 \; 7\}$

$\{2 \; 3 \; 7\}$

$\{1 \; 2 \; 3 \; 7\}$

↓

2^3

contribution in ans: $\Rightarrow$ A[i] * (1<<i);

↳ 2^i

```
int solve (int [] A) {
    int ans = 0;
    int n = A.length;
    Arrays.sort (A);            → nlogn
    for (int i=0; i<n; i++) {   ⎫
        ans += A[i] * (1<<i);   ⎬ n
    }                           ⎭
    return ans;
}
```

itr

$\Rightarrow n\log n + \cancel{n}$

TC: $O(n\log n)$

Doubts

```
int solve (int [] A) {
    int ans = 0;
    int n = A.length;
    Arrays.sort (A);

    for (int i=0; i<n; i++) {
        ans += A[i] * (1<<i);
    }
    return ans;
}
```

A = [3 2 4]

[2 3 4]
 0 1 2

i	freq	
0	1	{2}
1	2	{3} {2 3}
2	4	{4} {2 4} {3 4} {2 3 4}

Set bit

A = 3

B = 5

$0\ \cancel{0}^{1}\ 0\ \cancel{0}^{1}\ 0\ 0\ 0$

40

$ans = 2^A + 2^B$

Reverse bits

	7	6	5	4	3	2	1	0
8bit :	0	0	0	0	0	0	1	1
ans :	1	1	0	0	0	0	0	0

if ith bit in n is on then set $32-i-1$ bit in ans.

$8-i-1$

take care of calculations

↳ (long)