If we want to find intersection of some array, it's in less than O(n)

1. [2, 3] → 1100
2. [0, 1, 2] → 0111
3. [1, 3] → 1001

To find intersection, take AND.

1100
0111
0100 → 2nd fruit is common

→ We cannot use for more fruits as int, long long has limit

---

2. no. of workers ≤ 5000. They can work for max 30 days. Find ~~intersection of working days~~. two workers that are best for job. You need to assign an important project to 2 workers. Find max. intersection days

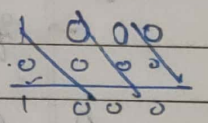
A
```
int n; cin >> n;
vector<int> masks(n, 0);
for (int i=0; i<n; ++i) {
  int num_workers;
  cin >> num_workers; int mask = 0;
  for (int j=0; i<num_workers; ++j)
  {
    int day; cin >> day;
    mask = (mask | (1<<day));
  }
  masks[i] = mask;
}
```

PTO →

~~#define ll long long~~

PAGE NO.
DATE: / /20

```
int n; cin>>n; int max_intersection = 0;
vector<int> masks(n,0);
for(int i=0; i<n; i++)
{
  int num_workers; cin >> num_workers;
  int mask=0;
  for(int j=0; j<num_workers; ++j)
  {
    int day; cin>>day;
    mask = (mask | (1<<day));   → to set the bits
  }                                  ↳ 1LL
  masks[i]= mask;
}
for(int i=0; i<n; ++i)
{
                i+1
for(int j=0; j<n; ++j)
{
  int intersection = (masks[i] & masks[j]);
  int common_days = __builtin_popcount(intersection);
  max_intersection = max(max_intersection, common_days);
}
}
```



1. __builtin_clz(a) → leading zeros
2. __builtin_ctz(a) → trailing zeros
3. XOR

8	→ N	8
7	→ 0	0
6	→ N+1	7
5	→ 1	1
4	← N →	4
3	← 0 →	0
2	← N+1 →	3
1	→ 1	1

Given x; it behaves like (x-1)%4+1

1^2^3^4 = 4

1^2^3 = 0

1^2 = 3

~~1^1~~

4. to get bitcount l = (int)log2(num) + 1;

Teacher's Signature

3. {4,3,4,4,4,5,5,3,3}

PAGE NO.
DATE: / /20

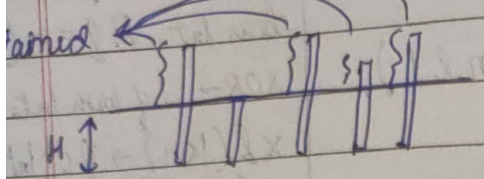
Property used → $x \wedge x = 0$

	set union A | B
	set intersection A & B
	set subtraction A & ~B
	set negation ~A
	set bit A |= 1<<bit
	clear bit A &= ~(1<<bit)
n & 1);	XOR → 1 if both bits are diff
	x & (1<<i) → if i^{th} bit of x is set
	0 if not set → !0 if set

~~Hashmaps~~ → Insertion, deletion, searching (O(1))

Advanced Binary Search

→ Mark the Woodcutter

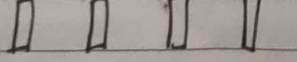


He needs atleast M wood
Find min. H at which he cuts.

Motive → max H for obtained wood >= M

```
bool isPossible (int a[], int n, int k, int mid)
{
  int wood = 0;
  for (auto ptr : a) if (ptr >= mid) wood += (ptr - mid);
  return wood >= k;
}
int main () {
  int n, k; cin >> n >> k; int ans = 0;
  int a[n] → input  int low = 0, high = INT_MAX, mid;
  while (low <= high)
  {
    mid = low + (high - low)/2;
    if (isPossible (a, n, k, mid) == true)
    { ans = mid, low = mid + 1; }
    else high = mid - 1; }
    cout << ans;
    return;
}
```

Aggressive cows



→ N stalls at x_0, x_1, x_2 coordinates

C cows to be placed such that min distance between them is as large as possible

Teacher's Signature

```
bool isPossible (int a[], int n, int c, int mid)
{
  int cows = 1; int last_pos = a[0];
  for (int i=1; i<n; i++)
  {   if (a[i] - last_pos >= mid) {cows++; last_pos = a[i];}
      if (cows == c) return true;
  }
  return false;
}
int main()
{
  int n; cin >> n; int c; cin >> c; int a[n]; cin → a
  sort(a, a+n);
  int low = 0, high = a[n-1], mid, ans = 0;
  while (low <= high)
  {
    mid = low + (high - low)/2;
    if (isPossible(a, n, c, mid) == true) ans = mid, low = mid+1;
    else high = mid-1;
  }
  cout << ans << endl;
}
```

Segmented Sieve ⇒ If low and high are till 10^{16} types

low = 6 high = 10

① Generate all primes from 0 to floor($\sqrt{high}$) → 2, 3

② Creat array of size (h − l + 1) i.e. (10 − 6 + 1) = 5

T	T	T	T	T

③ mark multiples of prime numbers generated

✗	T	✗	✗	✗
6	7	8	9	10

→ 7 is prime left in (6 to 10)

Teacher's Signature

DATE: / /20

Bit Manipulation

Q There is a N bit number, and a K & Z given such that
$Z = (X \ll 0) \wedge (X \ll 1) \wedge (X \ll 2) \ldots \wedge (X \ll (K-1))$
Find X.

A $X = 1010$ $K = 3$

```
  1 0 1 0
  1 0 1 0
  1 0 0 0
  1 1 0 1 1 0  → Z
```

$Z[i] = x[i]$ ('last $(k-1)$ ~~bits~~ bits of X)
$\Rightarrow Z[i]$ ('last $(k-1)$ bits of $x = x[i]$')

Q $n+i = n \wedge i$ Find such # $0 <= i <= n$
~~$\Rightarrow n \& i = 0$~~

A $n+i = n \wedge i$
$\Rightarrow n \& i = 0$
$\Rightarrow$ n में जहाँ 0 है i में वहां 0/1 हो सकते है।

x = no. of 0 in Binary rep of n
Ans = 2^x

Q Given an array, Find XOR of ~~all~~ elements of all subarrays possible.

A If i^{th} index $\Rightarrow$ its count in subarrays = $(i+1) * (N-i)$
we can get count of all elements
if even count XOR $\rightarrow 0$
if odd count XOR them once.

$[3, 5, 8, 3] \rightarrow$ subarray

3	5	8	3
3,5	5,8	8,3	
3,5,8		5,8,3	
		3,5,8,3	

count of 3 = (1) * (4) = 4
5 = (2) * (3) = 6
8 = (3) * (2) = 6
3 = (4) * (1) = 4

Teacher's Signature

Q- Given array, find sum of ^ of all 2 pairs possible.

A | 2 | 3 | 5 | Ans = (2^3) + (2^5) + (3^5)

We will go bit by bit

Suppose 0th bit of the numbers
→ a numbers have 0 at the psn
→ b numbers have 1 at the psn

Their contribution = (a * b)
Why? $^{a}C_1 . {}^{b}C_1$ ways of choosing a 1&0 so that XOR is 1.
Only choosing such configuration helps as 1^1 = 0^0 = 0.
∴ for each bit ⇒ a * b * pow(2, i)