Today's Content :

- → % Basics
- → Modular arthimetic

A % B = { Remainder }

12 % 5 = 2

a * b = a + a + -- a

a % b =

Dividend = Divisor * Quotient + Remainder

A % B = From A, keep subtracting B until value < B

30 % 7 = (30 - 7) : (23 - 7) : (16 - 7) : (9 - 7) : 2 < 7

40 % 7 = (40 - 7) : (33 - 7) : (26 - 7) : (19 - 7) : (12 - 7) : 5 < 7

5 % 5 = (5 - 5) : (0 < 5)

// Why do we use Mod? // To limit our range

$-\infty$, ∞ } → %10 → [0, 9]

a % M → [0, M-1]

Modular Arthmetic

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

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

($M \% M = 0$; $(a \% M) \in [0, M-1]$, so $(a \% M) \% M$ changes nothing)

$(a + M) \% M = (a) \% M$

c) $(a * b) \% M = (a \% M * b \% M) \% M$

d) $(a - b) \% M = ((a \% M - b \% M) + M) \% M$

$a = 10, b = 8, M = 9$

$(2) \% 9$

2

$= (1 - 8)$

$= (-7)$

$(2) \% 9$

2

$= (1 - 8 + 9) \% 9$

$2 = (2) \% 9$

$a = 7, b = 4, M = 8$

$(7 - 4) \% 8$

3

$(7 - 4 + 8) \% 8$

$(3 + 8) \% 8$

$(11) \% 8$

3

Q1) Given N +ve array elements, calculate no of pairs (i, j) such that $(arr[i] + arr[j]) \% M = 0$

Note: $i != j$ and { pair (i, j) is same as (j, i) } → 1 time

	0	1	2	3	4	5	
ar[6] =	4	7	6	5	8	3	M = 3

pairs:

i	j	ar[i] + ar[j]	
0	3	4 + 5 → 9 % 3 == 0 ✓	5 pairs
0	4	4 + 8 → 12 % 3 == 0 ✓	
1	3	7 + 5 → 12 % 3 == 0 ✓	
1	4	7 + 8 → 15 % 3 == 0 ✓	
2	5	6 + 3 → 9 % 3 == 0 ✓	
0	1	4 + 7 → 11 % 3 == 0 ? x	

Brute Force:

→ check all pairs

```
c=0
for(i=0; i<N; i=i+1){
    for(j=i+1; j<N; j=j+1){
        if( (ar[i] + ar[j]) % M == 0) { c++; }
    }
}
```

TC: $O(N^2)$ SC: $O(1)$

	0	1	2	3	4	5	6	
ar[7] =	[13	14	22	3	32	19	16]	M = 4

pairs?

i	j	ar[i] + ar[j] % M	
0	3	13 + 3 → 16 % 4 == 0 ✓	→ 4 pairs
0	5	13 + 19 → 32 % 4 == 0 ✓	
1	2	14 + 22 → 36 % 4 == 0 ✓	
4	6	32 + 16 → 48 % 4 == 0 ✓	

Idea:

$$(a + b) \% M = 0$$

$$\rightarrow (a \% M + b \% M) \% M = 0 \quad (b \% M \in [0, M-1])$$

- (1, M-1) : Consider only b values $b \% M = M-1$
- (2, M-2) : Consider only b values $b \% M = M-2$
- (3, M-3) : Consider only b values $b \% M = M-3$
- 0 — 0 : (Both are same)

if (M % 2 == 0) → M/2 — M/2 : (Both are same)

Eg:

	0	1	2	3	4	5	6	7	8	9	10	11
ar[12] =	6	7	5	11	19	20	9	15	14	13	12	23

M = 5

	0	1	2	3	4	5	6	7	8	9	10	11
ar[12] =	1	2	0	1	4	0	4	0	4	3	2	3

(2, 5), (2, 7), (5, 7) → How many ways to pick 2 from 3 → $^3C_2$ → 3

freq[0-4] :	0	1	2	3	4
freq :	3	2	2	2	3

6 pairs

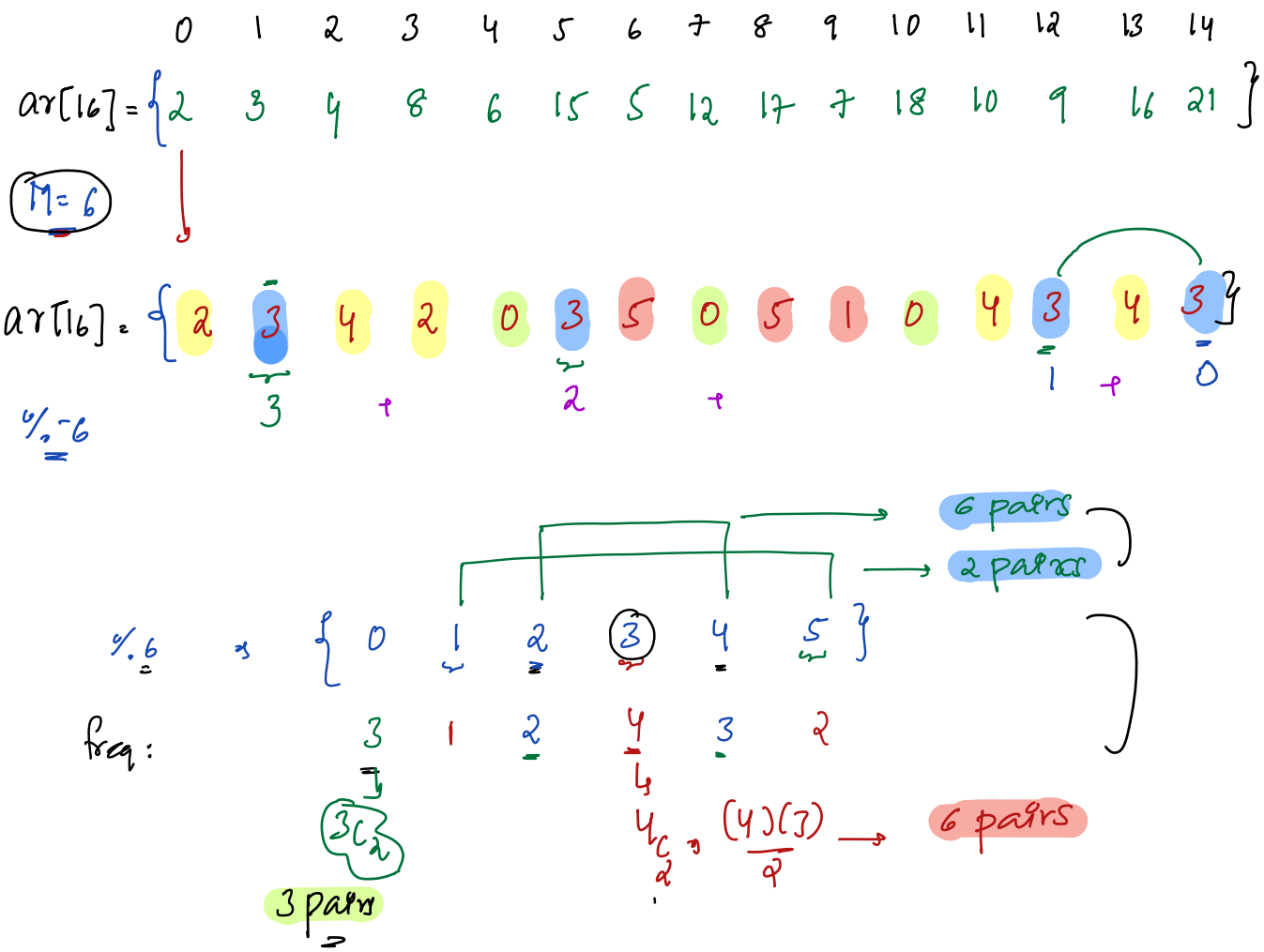
4 pairs

$\frac{(3)(3-1)}{2}$ → 3

Total = 13 pairs

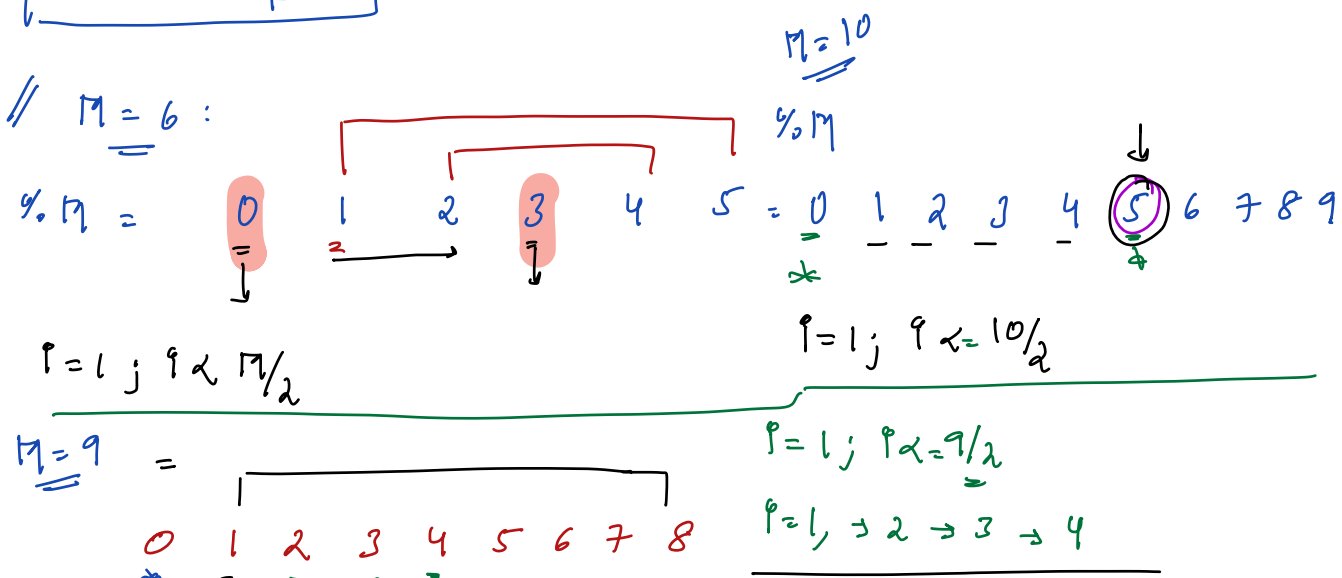
$$^NC_2 = \frac{(N)(N-1)}{2}$$

↳ how many ways we can pick 2 things from N things



Total = 17 pairs

// M = 6 :



Given ar[N] & m

SC

hashmap<int, int> hm;

int freq[m] : O(m)

→ freq [0, m-1]

hashmap :

↳ O(min(N, m))

hashmap better

// insert all freq in hm

```
for(i=0; i<N; i++){
    int x = ar[i]%m
    if(x is in hm){ // increase freq +1
        hm[x]++
    }
    else{ // insert(x, 1)
        hm.add(x,1)
    }
}
```

```
c = 0;
x = hm[0];
c = c + (x)(x-1)/2
if(m%2==0){
    x = hm[m/2]
    c = c + (x)(x-1)/2
}
for(i=1; i<(m+1)/2; i=i+1){
    c = c + hm[i] * hm[m-i]
}
return c;
```

m = 10 → (m+1)/2 → 5

i<5 → [1 2 3 4]

m = 9 → (m+1)/2 → 5

i<5 → [1 2 3 4]

TC: (N + m/2 + 1) → O(N+m)

// Space Complexity

N > M : {0, M-1} → O(M)

M = 100 →

N = 10

{0, M-1} → O(N)

→ SC : O(min(N, M))

10:45 break → & more question
→ & more concept

Q) Given N distinct ar[] elements, where $0 <= ar[i] <= N-1$

Replace every ar[i] = ar[ar[i]] SC:O(1)

- N Elements
- All Elements are from [0, N-1] have to present

```
        0  1  2  3  4
ar[5] = {3, 2, 4, 1, 0}
arr[] -> [1  4  0  2  3]
```

ar[0] = ar[ar[0]] = ar[3] = 1

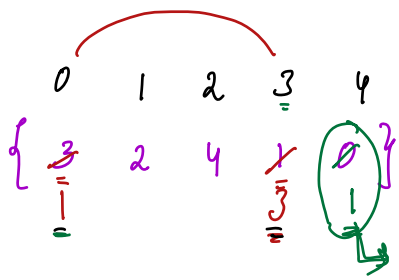
ar[1] = ar[ar[1]] = ar[2] = 4

ar[2] = ar[ar[2]] = ar[4] = 0

ar[3] = ar[ar[3]] = ar[1] = 2

ar[4] = ar[ar[4]] = ar[0] = 3

// Swap:



ar[0] = ar[ar[0]] = ar[3]

ar[4] = ar[ar[4]] = ar[0]

```
        0  1  2  3  4  5  6
ar[7] = {3, 1, 4, 6, 5, 0, 2}
ar[7] = 6  1  5  2  0  3  4
```

```
        0  1  2  3  4  5  6
ar[7] = {1, 6, 3, 5, 4, 2, 0}
ar[7] = 6  0  5  2  4  3  1
```

Swap:

- If we swap we loose
- Old data

// Idea: → No Swapping

→ No Extra Space

→ (At ar[i] we need to store both old & new data)

// Big-Bang:

n	Day	Hour
23	0	23
40	1	16
100	4	4
125	5	5
200	8	8

n/24 → Quotient → days

n%24 → Remainder → time

↦ /n : old

n ↳ %n : new

new } TODO

old

n	n/24	n%24

n ↦ /n old

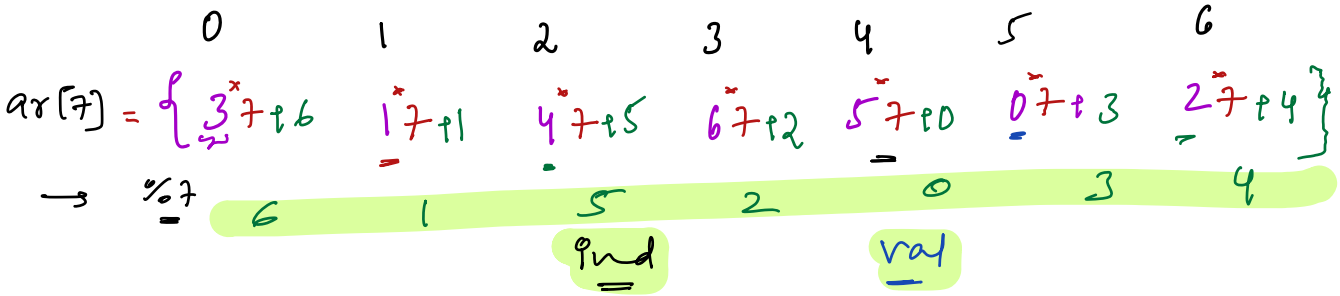
↳ %n : new

→ n = old*n + new

↦ /n = old

↳ %n = new

// ar[i] = old*(N) + new



ar[0] = ar[ar[0]] ar[0]/7 = 3 ar[3]/7 = 6, add at ar[0]

find val

ar[1] = ar[ar[1]] ar[1]/7 = 1 ar[1]/7 = 1, add at ar[1]

ar[2] = ar[ar[2]] ar[2]/7 = 4 ar[4]/7 = 5, add at ar[2]

ar[3] = ar[ar[3]] ar[3]/7 = 6 ar[6]/7 = 2, add at[3]

ar[4] = ar[ar[4]] ar[4]/7 = 5 ar[5]/7 = 0 add ar[4]

ar[5] = ar[ar[5]] ar[5]/7 = 0 ar[0]/7 = 3 add ar[5]

ar[6] = ar[ar[6]] ar[6]/7 = 2 ar[2]/7 = 4 add ar[6]

//

Pseudo code

1)
```
for(i=0; i<N; i=i+1){
    ar[i] = ar[i]*N
}
```
ar[i] = old * N

2)
```
for(i=0; i<N; i=i+1){
    // ar[i] = ar[ar[i]]
    ind = ar[i]/N
    val = ar[ind]/N
    ar[i] += val
}
```
ar[i] = old * N + new

3)
```
for(i=0; i<N; i=i+1){
    ar[i] = ar[i]%N
}
```
ar[i] = new

TC: O(N)

SC: O(1)

→ $(a/b) \% M = (a\%M / b\%M)$

→ $(a=3, b=2, M=2)$ $(0)/(0)$

$(2/2) \% 2 = 1$

→ $(a/b) \% M \rightarrow (a * b^{-1}) \% M$, $(a\%M * b^{-1}\%M) \% M$

Inverse modulo

→ Given a, M, $(a^{-1}\%M)$ is exists if $gcd(a, M) = 1$

↳ proof:

↳ {greatest common divisor}

// $gcd(a, M) = 1$

given a, M, $b = (a^{-1}) \% M \rightarrow [1, M-1]$

$(a * b) \% M = 1$

// What's value of $b = [1, M-1]$

// $a = 7$, $M = 10$, $gcd(a, M) = 1$ → $a^{-1}\%M = 3$

$b = 9$

$b = (1 * 7) \% 10 \neq 1$

$b = (2 * 7) \% 10 \neq 1$

$b = (3 * 7) \% 10 == 1$

$b = 4$

$b = 5$

$b = 6$

$b = 7$

$b = 8$

// $a = 10, \quad m = 9 \Rightarrow \gcd(a, m) = 1$

$b = a^{-1} \% m \Rightarrow [1 \quad 8] \Rightarrow a^{-1} \% m = 1$

$(10 * 1) \% 9 = 1$

$a = 6 \quad m = 10 \Rightarrow \gcd(a, m) = 2, \quad b = (a^{-1}) \% m \Rightarrow [1, m-1]$

$b = (1 * 6) \% 10 \neq 1$
$(2 * 6) \% 10 \neq 1$
$(3 * 6) \% 10 \neq 1$
$(4 * 5) \% 10 \neq 1$
$(5 * 6) \% 10 \neq 1$
$(6 * 6) \% = 10$
$(7 * 6) \% 10 \neq 1$
$(8 * 6) \% 10 \neq 1$
$(9 * 6) \% 10 \neq 1$

// no invers modulu

// Pseudo code

Given A, m, $\gcd(a, m) = 1$, $b = a^{-1} \% m \rightarrow [1, m-1]$

```
i = 1; i < m; i++){
    if ( (a * i) % m == 1) {
        return i
    }
}
```

TI: $O(M)$

SI: $O(1)$

// Extended Eulerian : $O(\log m)$: C (Very Very Tricky)

If m is a prime number :

{Fermat's little Theorm}

$\rightarrow (a^{m-1}) \% m = 1$

// Multiply $(a^{-1}) \% m$ on both sides

$= (a^{m-1} * a^{-1}) \% m = (a^{-1}) \% m$

$= (a^{m-2}) \% m = (a^{-1}) \% m$

using power function $\rightarrow$ $(\log(m))$

$a = 8, \quad m = 11 \rightarrow \gcd(a, m) = 1, \quad m$ is a prime ✓

$(a^{-1}) \% m = (a^{m-2}) \% m = (8^{9}) \% 11 = (23)^{9} \% 11$

$(a^{-1}) \% m = 7 \quad \rightarrow (2^{27}) \% 11 = 7$

$((a) * (a^{-1}) \% m) \% m = (7 * 8) \% 11 \rightarrow (56) \% 11 \rightarrow 1$

Doubts → Storing 3 numbers in single variable

$$n = a_0 + a_1 N + a_2 N^2$$

$n / N^2 : a_2$

$(n \% N^2) / N = a_1$

$(n \% N) = a_0$

// $a_0, a_1, a_2 < (N)$

$a_0\ a_1\ a_2 >= 0$