Todays Content:

- % Basics
- % Arthmetics
- Problems
  a) A%.M = B%.M

Very Imp Interview
  b) No: of pairs
  c) Replace A[i] with A[A[i]]

Sunday Problem Solving Session?
- 10am → 1pm
- recordings available:

Inverse Modulus & optimal content & Doubt Session

%.

14 % 5 = 4

40 % 7 = 5

A % B = remainder when A is divided by B

{ Input } % B = { min 0 , max B-1 }

Modular Arithmetic:

% → [+, -, *, /]

[[0, m-1] [0, m-1]] % m = [0, m-1]

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

[0, m-1]

$$(a \% m) \% m = a \% m$$

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

$$(a - b) \% m = (a \% m - b \% m) \% m$$ In Java fails, Python works

a b m

13 9 5 = (13 % 5 - 9 % 5) % 5

= (13 - 9) % 5 = (3 - 4) % 5

= (4) % 5 = (-1) % 5 → In python: 4

4 ↳ In Java: -1

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

$$(a - b) \% m = (a \% m - b \% m + m) \% m$$ will hold in all languages

a b m

13 9 5 = (13 % 5 - 9 % 5 + 5) % 5

= (3 - 4 + 5) % 5

4 = 4 % 5 = 4

Q: Given A, B & A > B find no: of m > 0 such that

$A \% m = B \% m$

Ex: A = 10 B = 4, ans = 4

m	A % m	B % m
1	10 % 1 =	4 % 1
2	10 % 2 =	4 % 2
3	10 % 3 =	4 % 3
4	10 % 4	4 % 4
5	10 % 5	4 % 5
6	10 % 6	4 % 6
7	10 % 7	4 % 7
8	10 % 8	4 % 8
9	10 % 9	4 % 9
10	10 % 10	4 % 10

T10 not matching

If Say m > A: A > B

A % m = A
B % m = B
not possible

Idea: Iterate from m = {1 → A}

```
c = 0
m = 1; m <= A; m++){
    if( A % m == B % m) { c = c+1}
}
```

TC: O(A)

Opt: $A \% m = B \% m$

$A \% m - B \% m = 0$

// Add m on both sides

$[A \% m - B \% m + m] = m$

// Apply %m on both sides

$[A \% m - B \% m + m] \% m = 0$

$(A - B) \% m = 0$:

$12 \% n = 0$

Find all n which satisfies

n = {1, 2, 3, 4, 6, 12}

Obs: n is all factrs of 12

M should be all factrs A − B

// Given N find all factrs: $\sqrt{N}$

TC: $\sqrt{A-B}$ SC: O(1)

(8) Given ar[N], M, calculate no of pairs i, j such that

$\{ar[i] + ar[j]\} \% M = 0$

Note: $i \neq j$ and pair(i, j) is same as pair(j, i)

Consider pair only once:

Eg:

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

i	j	ar[i]	ar[j]	{ar[i] + ar[j]} % M
0	3	4	5	9 % 3 == 0 ✓
0	4	4	5	9 % 3 == 0 ✓
1	3	7	5	12 % 3 == 0 ✓
1	4	7	5	12 % 3 == 0 ✓
2	5	6	3	9 % 3 == 0 ✓

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

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

Idea: Check all pairs TC: O(N²) SC: O(1)

```
c = 0
i = 0 → n
    j = i+1 → n
        if( {ar[i] + ar[j]} % m == 0) { c = c+1 }
return c;
```

Idea2: $(ar[i] + ar[j]) \% M = 0$

Expand: $(ar[i] \% M + ar[j] \% M) \% M = 0$

$ar[i] \% M$ → $[0, M-1]$; $ar[j] \% M$ → $[0, M-1]$

$[0 + 0] \% M = 0$

$[2 + M-2] \% M = 0$

$[3 + M-3] \% M = 0$

$[4 + M-4] \% M = 0$

Eg:

$A = 13, M = 4, B \% M = 3$

$(A + B) \% M = 0$

$(A \% M + B \% M) \% M = 0$

$[1 + 3] \% 4 = 0$

$A = 24, M = 8, B \% M = 0$

$[A + B] \% 8 = 0$

$[A \% 8 + B \% 8] \% 8 = 0$ ($B \% 8$ → $\{0-7\}$)

$[0 + 0] \% 8 = 0$

$A = 7, M = 3, B \% M = 2$

$[A + B] \% 3 = 0$

$[A \% 3 + B \% 3] \% 3 = 0$

$[1 + 2] \% 3 = 0$

$A = 25, M = 10, B \% M = 5$

$[A + B] \% 10 = 0$

$[A \% 10 + B \% 10] \% 10 = 0$ ($B \% 10$ → $\{0-9\}$)

$[5 + 5] \% 10 = 0$

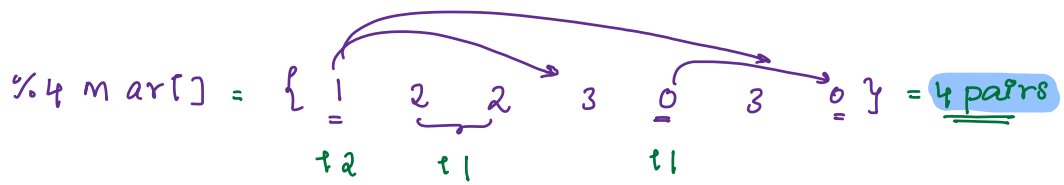
Idea: $(ar[i] + ar[j]) \% m = 0$

$[ar[i] \% m + ar[j] \% m] \% m = 0$

Edge Case

0	0
1	m-1
2	m-2
3	m-3
⋮	
k	m-k

Eg: $ar[7] = \{13 \quad 14 \quad 22 \quad 3 \quad 32 \quad 19 \quad 16\} \quad m = 4$

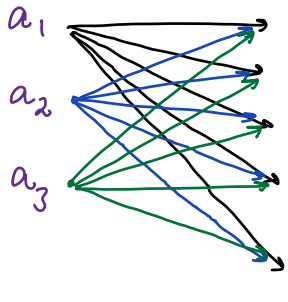


Q): Given ar[8], find no: of pair $[ar[i] + ar[j]] \% 5 = 0$

↳ Inf: Say there are 3 elements $ar[i] \% 5 = 1$

Say there are 5 elements $ar[i] \% 5 = 4$

$ar[i] \% 5 = 1$ $\quad$ $ar[i] \% 5 = 4$ } 15 pairs



Idea: We simply need freq of all remainders

ar[], M = 10,

ar[ ] =

29 11 21 17 2 5 4 6 23 13 26 14 18 15 30 35 50 20 40 9

cnt[10] =

0	1	2	3	4	5	6	7	8	9
4	2	1	2	2	3	2	1	1	2

30 11 2 23 4 5 6 17 18 29
50 21 13 14 15 26 9
20 35
40

ar[], M = 8

M = 10: [ar[i] % 10 + ar[j] % 10] % 10 = 0

Edge Can Happen for

1: cnt[0]

2: cnt[M/2]

cnt[0] * cnt[0] = 16 pairs *

cnt[1] * cnt[9] = 4 pairs

cnt[2] * cnt[8] = 1 pair

cnt[3] * cnt[7] = 2 pairs

cnt[4] * cnt[6] = 4 pairs

cnt[5] * cnt[5] = 9 pairs *

Handle:

Edge Cases:

rem = 0: 30 50 20 40 (+3 +2 +1 0) pair = 6, a = 4 → $\frac{[4][3]}{2}$ → 6

rem = 5: 15 15 35 (+2 +1 0) pair = 3, a = 3 → $\frac{[3][2]}{2}$ → 3

Note: Given a times, no: of ways to pick 2 items = $\frac{(a)(a-1)}{2}$

```
int countPairs(int arr[], int n, int m){   TC: O(N+m)  SC: O(m)

    // Step 1: Store freq of remainder %m  ──> cnt[m] ✓
                                           ──> Hashmap : TODO

    cnt[m] = 0
    i = 0; i < n; i++){                    ] TC: N
        cnt[arr[i] % m]++                  ] arr
    }

    // Step 2: Handle Edge Case 1: O(1)
        ans = 0
        a = cnt[0] // no: of elements with remainder = 0
        ans = ans + (a)(a-1)/2

    Handle Edge Case 2:
        if( m%2 == 0){ // if m is even only then
            b = cnt[m/2]                    cnt[m/2]  cnt[m/2]
            ans = ans + (b)(b-1)/2          Case occurs
        }

    // Step: 3 Get all remaining pairs
        i = 1, j = m-1   TC: O(m)
        while( i < j ){
            ans = ans + cnt[i] * cnt[j]
            i = i+1      #elements  #element
            j = j-1      with rem i  with rem j
        }
        return ans;
}
```

Q) Given an ar[N], which contain all elements from [0, N-1]

Replace ar[i] → ar[ar[i]] → {Google/Amazon} SC: O(1)

Note: modify given input ar[]

N = 5

	0	1	2	3	4
ar[5] =	3	2	4	1	0
	ar[ar[0]]	ar[ar[1]]	ar[ar[2]]	ar[ar[3]]	ar[ar[4]]
	ar[3]	ar[2]	ar[4]	ar[1]	ar[0]
after modi ar[] ar[5] =	1	4	0	2	3

Eg:

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

Eg:

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

With Extra Space:

```
int[] replace(int ar[], int n){ TC: O(N) SC: O(N)
    int b[n];
    for(i=0; i<n; i++){
        b[i] = ar[ar[i]]
    }
    copy b[] → ar[]
    return ar[]
}
```

without space

ar[i] = ar[ar[i]] * wrong

	0	1	2	3	4	5	6
Qn: ar[7] =	~~3~~ 6	~~4~~1	~~4~~5	~~6~~2	~~8~~0	~~0~~6	~~2~~5
=	6	1	5	2	0	6	5

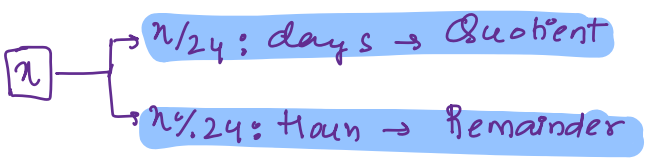
obs: We need to store both old & new data in same arr[] ?

⟶

Bing Bang: [Universe started]

Day: 0

Hours:	Day	Time
25 ⟶	1	1 hr
50 ⟶	2	2 hrs
100 ⟶	4	4 hrs
130 ⟶	5	10 hrs
n: ⟶	n/24	n%24



old data [0, n-1] new-data [0, n-1]

Obs: Say $x = [x\ n] \rightarrow x/n = 0$

$x\%n = x$

Say: $ar[i] = old * n + new$ { $ar[i]/n = old$; $ar[i]\%n = new$ } version-1

$ar[i]/n = \{old * n + new\}/n = \frac{old * n}{n} + \frac{new}{n} = old + 0$

$ar[i]\%n = \{old * n + new\}\%n = \{[old * n]\%n + [new]\%n\}\%n$

$= 0 + [new\%n]\%n = new\%n = new$ ($u\%[0, n-1]$)

$ar[i] = new * n + old$ { $ar[i]/n = new$; $ar[i]\%n = old$ } version-2 TODO

$ar[i] = old * n + new$

	0	1	2	3	4	5	6
N = 7							
Eg: $ar[7] =$	$1*7+4$	$4*7+6$	$3*7+5$	$5*7+2$	$6*7+0$	$2*7+3$	$0*7+1$

newdata

$ar[0] = ar[ar[0]/7] = ar[1]/7 = 4*7/7 = 4$

$ar[1] = ar[ar[1]/7] = ar[4]/7 = \frac{6*7}{7} = 6$

$ar[2] = ar[ar[2]/7] = ar[3]/7 = \frac{5*7}{7} = 5$

$ar[3] = ar[ar[3]/7] = ar[5]/7 = \frac{2*7}{7} = 2$

$ar[4] = ar[ar[4]/7] = ar[6]/7 = \frac{0*7}{7} = 0$

$ar[5] = ar[ar[5]/7] = ar[2]/7 = \frac{3*7+5}{7} = 3$

$ar[6] = ar[ar[6]/7] = ar[0]/7 = \frac{1*7+4}{7} = 1$

	0	1	2	3	4	5	6
N = 7							
Eg: $ar[7] =$	$1*7+4$	$4*7+6$	$3*7+5$	$5*7+2$	$6*7+0$	$2*7+3$	$0*7+1$

Apply %7 to extract only new data

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

```
replace(int ar[N]) { TC: O(N) SC: O(1)

    // Step1: ar[i] = old*n        Storing 2 things at same place
    i=0; i<n; i++){
        ar[i] = ar[i]*n
    }
    // Step2: ar[i] = old*n + new
    i=0; i<n; i++){
        ar[i] += { ar[ar[i]/n]/n }
    }
    // Step3: ar[i] = new data
    i=0; i<n; i++){
        ar[i] = ar[i]%n
    }
    return ar
}
```

Sunday: 10:am → as long as it takes

10am → 10:20am: Invcm module

10:20 → All doubts:

→ arrays { 2

→ bits → 1