Pigeonhole Principle

- Assume we have N=5 boxes and M=6 apples
- Distribute apples in boxes in whatever way
- There must be 1 box with at least 2 apples
- If you want to avoid duplicate:
  - Put 1 apple per box. Remaining is 1 apple.
  - To put it, one box will have 2 apples
- What if we have 13 apples?
  - 1 box will have at least $\lceil 13/5 \rceil = 3$ apples
- Generally: $\lceil N/M \rceil$ per a box.

Pigeonhole Principle: Example

- **Prove**: Among any N positive integers, there exists 2 whose difference is divisible by N-1.
- Recall: |A-B| % X = 0 IFF A%X = B%X
- So, let's compute % N-1?
  - Then we have N-1 values, each in range [0 - N-2]
  - But, we have N numbers, then at least one mode will be duplicate
  - Given that 2 numbers at least has same % N-1
  - Then, their difference must be divisible by N-1
- **Always** relate % with Pigeonhole

Pigeonhole Principle: Example

- Case for previous problem:
- Let A represents array of numbers
- N = 5 and A = {2, 3, 5, 7, 8}
- Compute A % 4 = {2, **3**, 1, **3**, 0}
- 5 numbers, with values [0 - 3].
- Pick 2 with same mode - 3 is **repeated** mode
- Then, (3, 7) are the answer

Pigeonhole Principle: Example

- Prove: For any N positive integers, the **sum** of **subset** of them is divisible N
- Compute Accum array % N
  - Accum[i] = {A[0] + A[1]...A[i]} % N
- If any Accum[i] = 0, we are done
- Otherwise, we have N values in [1 - N-1]
- Then 2 positions will have **same mode**
- Then getting numbers between them is answer

Pigeonhole Principle: Example

- Let A represents array of numbers
- N = 5 and A = {2, 4, 8 , 2 , 7}
- Accumulate: B = {2, 6, 14, 16, 23}
- Mode 5: C = {2, 1, 4, 1 , 3}
- Any zeros? No..remaining 4 values spread on 5 values...one of them must be repeated
  - if yes then A[0]+A[1]...A[i] where C[i] % N = 0
- 2nd and 4th have mode 1
- Then range from 3rd till 4th is answer: 8, 2

Pigeonhole and Competitions

- Most of time it helps in **proving**, rather than a technique to apply
  - Read many problem examples (web/books) + proofs
- In some problems, it can be the major trick
- Sometimes comes with Modular Arithmetic
- Some facts in graph:
  - A Path of M nodes (M > N) must have a node used more than once
  - A cycle of Length M (M > N nodes), must be composed of cycles each of Length <= N
  - Every graph contains two vertices of same degree

Powers tower % M

- Let's compute: 2^3^4^5^6^7^8 % 56
  - We can solve it using Euler theorem
- let's simplify it, compute 2^X % 56
  - where x is very large, e.g. x = 3^4^5^6^7^8
- Imagine we compute 2^i%M for i [0 - OO]
  - We know we have M mod values: [0 - M-1]
  - Pigeonhole: values repeat in **maximum M + 1** iterations
  - Then computing X should have **same value** as one of the first powers in range [0-M-1]
  - But which 2^i%M correspond to 2^x%M ?
- Let's simulate it

Powers tower % M

i	0	1	2	3	4	5	6	7
$2^i \% 56$	1	2	4	**8**	16	32	**8**	16

- 2^6 is same as 2^3. Then 2^7 must = 2^4...etc
- {8,16,32} is **cycle** and {1,2,4} is **precycle**
  - Let length of the cycle be **L**, and length of precycle be **P**
- Given some X, we can compute its i position
  - i = P + (L + **X % L** - P % L) % L   [if X > P]
- Then?
  - Solve subproblem X % L in same manner
  - Then compute 2^i % M