Brute Force Algorithm (Yunhao Zhang)

I. 3-step Approach to Interview

step1 try brute force
step2 justify whether brute force is enough
step3 if not, try advanced algorithms

today’s class

II. A Starting Problem

codeforces link: https://codeforces.com/problemset/problem/681/B

Input: an integer \( n \) \( (1 \leq n \leq 10^9) \)

Output: YES if there exists non-negative integers \( a, b, c \) that \( a \times 1234567 + b \times 123456 + c \times 1234 = n \)

NO o.w.

step1 Solution: for-loop enumeration style of brute force

```
for (int a=0; a \leq \frac{n}{1234567}; a++)
  for (int b=0; b \leq \frac{n}{123456}; b++)
    if ((n-a*1234567-b*123456) % 1234 == 0)
      return true
    return false
```
Notice! Tricky!

① $10^9$ is a large number and using `long long` is safer than `int` for arithmetic calculations.

② if, in the problem, $c$ is positive integer instead of non-negative integer, one needs to check $c \neq 0$ in the inner if-statement.

Step2 justification

Why the brute force algorithm is sufficient?

In the algorithm, there are 2 loops:

- `int a` from 1 to $\frac{n}{1234567}$
- `int b` from 1 to $\frac{n}{123456}$

since $n \leq 10^9$, the maximum # loop is

$$\frac{10^9}{1234567} \times \frac{10^9}{123456} \approx 810 \times 8100 = 6561000$$

Experience:

- if $N \leq 10^6$, O(N) algorithms usually run within 1 second perfectly
- if $N \leq 10^7$, a simple O(N) loop usually run within 1 second perfectly

Conclusion: since $6561000 \leq 10^7$, this brute force solution is efficient enough and can run within 1 second.
Homework

codeforces link: https://codeforces.com/problemset/problem/727/A

For this problem, repeat step 1 and step 2, namely try brute force algorithm and justify that brute force is efficient enough.