北大合宿 2019 Day2 F: MOD Rush

原案: tempura
問題文: tsutaj
解答: idsigma・tempura・tsutaj・tubuann
解説: tsutaj

2019 年 7 月 15 日

問題

MOD Rush

- 長さ N の正の整数列 A と、長さ M の正の整数列 B が与えられる
- 全ての (i, j) $(1 \leq i \leq N, 1 \leq j \leq M)$ に対して、A_i を B_j で割ったあまりを計算し、その和を求めよ
- 制約
  - $1 \leq N, M \leq 2 \times 10^5$
  - $1 \leq A_i, B_j \leq 2 \times 10^5$

想定誤解法

$O(NM) \to$ Time Limit Exceeded

```cpp
#include <cstdio>
int main() {
    int N, M, A[200010], B[200010]; scanf("%d%d", &N, &M);
    for(int i=0; i<N; i++) scanf("%d", &A[i]);
    for(int i=0; i<M; i++) scanf("%d", &B[i]);

    long long int ans = 0;
    for(int i=0; i<N; i++) for(int j=0; j<M; j++) ans += A[i] % B[j];
    printf("%lld\n", ans);
    return 0;
}
```

アプローチ

- 剰余のまま考察するのは扱いにくいので、問題を言い換えよう
- そもそも剰余とは？

アプローチ

- 剰余のまま考察するのは扱いにくいので、問題を言い換えよう
- そもそも剰余とは？
  - X を Y で割ったあまりを求めたいとする
  - $X = pY + q$ の形で表すことができ、この時の q が剰余
    - $p, q \in \mathbb{N} \cup \{0\}$
    - $0 \leq q < Y$
  - 商が求められれば、X から商を Y 倍したものを引いて剰余が出せる

アプローチ

- 剰余のまま考察するのは扱いにくいので、問題を言い換えよう
- そもそも剰余とは？
  - X を Y で割ったあまりを求めたいとする
  - $X = pY + q$ の形で表すことができ、この時の q が剰余
    - $p, q \in \mathbb{N} \cup \{0\}$
    - $0 \leq q < Y$
  - 商が求められれば、X から商を Y 倍したものを引いて剰余が出せる
- よって元の問題は次のように言い換えることができる

アプローチ

- 剰余のまま考察するのは扱いにくいので、問題を言い換えよう
- そもそも剰余とは？
  - X を Y で割ったあまりを求めたいとする
  - $X = pY + q$ の形で表すことができ、この時の q が剰余
    - $p, q \in \mathbb{N} \cup \{0\}$
    - $0 \leq q < Y$
  - 商が求められれば、X から商を Y 倍したものを引いて剰余が出せる
- よって元の問題は次のように言い換えることができる

MOD Rush の言い換え

正の整数列 A, B に対し、以下の操作を高速に実行せよ

- $S \leftarrow M \sum_i A_i$ とおく
- 各 (i, j) に対して、$p = \lfloor \frac{A_i}{B_j} \rfloor$ を求め、S から $p \times B_j$ を引く
- 最終的に得られた S を出力

アプローチ

- どうすれば商を高速に求められるか？

アプローチ

- どうすれば商を高速に求められるか？
- B_j をひとつ選びとったとき、B_j で割った商が p になるような A_i の範囲は連続した区間になり、商の値によらず区間の長さは B_j
  - 商が p であるような A_i は、$p \times B_j \leq A_i < (p+1) \times B_j$ を満たすもの
  - $P_i :=$「A 内に登場した数の中で値が i 以下であるものがいくつあるか」という配列を用意しておけば、範囲内にあるものは $O(1)$ で数えることができる (いわゆる累積和)

アプローチ

- どうすれば商を高速に求められるか？
- B_j をひとつ選びとったとき、B_j で割った商が p になるような A_i の範囲は連続した区間になり、商の値によらず区間の長さは B_j
  - 商が p であるような A_i は、$p \times B_j \leq A_i < (p+1) \times B_j$ を満たすもの
  - $P_i :=$「A 内に登場した数の中で値が i 以下であるものがいくつあるか」という配列を用意しておけば、範囲内にあるものは $O(1)$ で数えることができる (いわゆる累積和)
- $X = \max A_i$ とおくと、試す区間の数は $\lceil \frac{X}{B_j} \rceil$ 個
- B_j が同じ値であればまとめて計算すれば良いことを考えると、B は相異なるとみなしてよい
- 試すべき区間の数の合計は最大 $\lceil \frac{X}{1} \rceil + \lceil \frac{X}{2} \rceil + \cdots + \lceil \frac{X}{\max B_i} \rceil$ 個

アプローチ

- 区間の数の合計は最大 $\lceil \frac{X}{1} \rceil + \lceil \frac{X}{2} \rceil + \cdots + \lceil \frac{X}{\max B_i} \rceil \approx X \log X$ 個
  - 調和級数と呼ばれるもの
- よって全ての (i, j) に対して商を求め、和にすることは高速に可能
- これを利用すると剰余の部分だけ残すことができ、元の問題も高速に処理可能 → Accepted

Writer 解・統計

- Writer 解
    - idsigma (C++・35 行・838 bytes)
    - tempura (C++・37 行・652 bytes)
    - tsutaj (C++・28 行・819 bytes)
    - tubuann (C++・63 行・1638 bytes)
- 統計
    - AC / tried: 27 / 67 (40.3 %)
    - First AC
        - On-site: Tampaku (93 min 54 sec)
        - On-line: ushitapunichiakun (8 min 16 sec)