1　包除原理

　数え上げのテクニックの一つ．集合 A が与えられ，その要素を変数とする述語たち $\mathcal{P} = \{P_1, \ldots, P_n\}$ を考える．$1 \leqslant i \leqslant n$ に対して A の部分集合 A_i を $A_i = \{a \in A \mid P_i(a)\}$ で定めるとき，$\left|\bigcup_{i=1}^n A_i\right|$ を求めるものである．$\Lambda_n = \{1, \ldots, n\}$ とする．

$$\left|\bigcup_{i=1}^n A_i\right| = \sum_{\emptyset \subset \Lambda \subseteq \Lambda_n} (-1)^{|\Lambda|-1} \cdot \left|\bigcap_{i \in \Lambda} A_i\right| . \tag{1}$$

$$= \sum_{j=1}^n (-1)^{j-1} \cdot \left(\sum_{|\Lambda|=j} \left|\bigcap_{i \in \Lambda} A_i\right|\right) . \tag{2}$$

Λ の要素数を固定したときに $\left|\bigcap_{i \in \Lambda} A_i\right|$ の総和を DP などで求められるなら，式 (2) を用いて計算することができる．また，特に $|\Lambda| = |\Lambda'| \implies \left|\bigcap_{i \in \Lambda} A_i\right| = \left|\bigcap_{i \in \Lambda'} A_i\right|$ であるなら，$|\Lambda| = j$ であるような Λ の代表元を Λ^j と書くことにして次のように変形できる．

$$\left|\bigcup_{i=1}^n A_i\right| = \sum_{j=1}^n (-1)^{j-1} \cdot {}_n C_j \cdot \left|\bigcap_{i \in \Lambda^j} A_i\right| . \tag{3}$$

式 (1) においては $\sum$ で足される項が $2^n - 1$ 個だったのに対し，式 (2)–(3) では n に減っていてうれしい．

　上の議論は，$\left|\bigcap_{i \in \Lambda} A_i\right|$ を計算するのが容易であることを前提としているが，逆に $\left|\bigcap_{i \in \Lambda} (A \setminus A_i)\right|$ の計算が容易な状況[*1]で $\left|\bigcap_{i=1}^n A_i\right|$ を求めたいときには以下のようにするとよい．

$$\bigcap_{i=1}^n A_i = A \setminus \left(\bigcup_{i=1}^n (A \setminus A_i)\right) .$$

すなわち，

$$\left|\bigcap_{i=1}^n A_i\right| = |A| - \left|\bigcup_{i=1}^n (A \setminus A_i)\right| . \tag{4}$$

右辺の第 2 項は，$A \setminus A_i$ を A_i と置き直すことで式 (1) の枠組みで求められる．

[*1] 満たす条件を決め打ちするよりも，満たさない条件を決め打ちした方が楽な場合．

1.1 証明

$\mathcal{P}$ の大きさ n に関する帰納法で示す．$n = 2$ の場合は Venn 図などから示される（省略）．$n < k$ で成り立っていることを仮定する．まず，$n = 2$ の場合の式を用いて次のように変形する．

$$\begin{aligned}\left|\bigcup_{i=1}^{k} A_i\right| &= \left|\left(\bigcup_{i=1}^{k-1} A_i\right) \cup A_k\right| \\ &= \left|\bigcup_{i=1}^{k-1} A_i\right| + |A_k| - \left|\left(\bigcup_{i=1}^{k-1} A_i\right) \cap A_k\right| \\ &= \left|\bigcup_{i=1}^{k-1} A_i\right| + |A_k| - \left|\bigcup_{i=1}^{k-1} (A_i \cap A_k)\right|\end{aligned}$$

さらに，$n = k - 1$ の場合の式を用いて右辺の各項は次のように変形できる．

$$\left|\bigcup_{i=1}^{k-1} A_i\right| = \sum_{\emptyset \subset \Lambda \subseteq \Lambda_{k-1}} (-1)^{|\Lambda|-1} \cdot \left|\bigcap_{i \in \Lambda} A_i\right|.$$

$$|A_k| = \sum_{\Lambda = \{k\}} (-1)^{1-1} \cdot \left|\bigcap_{i \in \Lambda} A_i\right|.$$

$$\begin{aligned}-\left|\bigcup_{i=1}^{k-1} (A_i \cap A_k)\right| &= -\sum_{\emptyset \subset \Lambda \subseteq \Lambda_{k-1}} (-1)^{|\Lambda|-1} \cdot \left|\bigcap_{i \in \Lambda} (A_i \cap A_k)\right| \\ &= -\sum_{\emptyset \subset \Lambda \subseteq \Lambda_{k-1}} (-1)^{|\Lambda|-1} \cdot \left|\left(\bigcap_{i \in \Lambda} A_i\right) \cap A_k\right| \\ &= \sum_{\emptyset \subset \Lambda \subseteq \Lambda_{k-1}} (-1)^{|\Lambda \cup \{k\}|-1} \cdot \left|\bigcap_{i \in \Lambda \cup \{k\}} A_i\right|.\end{aligned}$$

第 1 項は A_k を含まない $k-1$ 個以下の積集合，第 2 項は A_k のみからなる集合，第 3 項は A_k を含む 2 個以上 k 個以下の積集合に関する式になっており，次のようにまとめることができる．

$$\left|\bigcup_{i=1}^{k} A_i\right| = \sum_{\emptyset \subset \Lambda \subseteq \Lambda_k} (-1)^{|\Lambda|-1} \cdot \left|\bigcap_{i \in \Lambda} A_i\right|. \quad \square$$

1.2 発展

1.2.1 高速ゼータ変換・高速 Möbius 変換

高速ゼータ変換は，集合 S の部分集合を引数に取る関数 f に対して $g(S) = \sum_{T \subseteq S} f(T)$ を $O(|S| \cdot 2^{|S|})$ 時間で求める．高速 Möbius 変換はその逆変換に相当し，$f(S) = \sum_{T \subseteq S} (-1)^{|S \setminus T|} \cdot g(T)$ を求める．

```
// fast zeta transformation
for (size_type i = 0; i < n; ++i)
  for (size_type j = 0; j < (1_zu << n); ++j)
    if (j >> i & 1_zu) dp[j] += dp[j ^ (1_zu << i)];
```

```
// fast Moebius transformation
for (size_type i = 0; i < n; ++i)
  for (size_type j = 0; j < (1_zu << n); ++j)
    if (j >> i & 1_zu) dp[j] -= dp[j ^ (1_zu << i)];
```

各 j について $\mathrm{dp}[j] = f(j)$ で初期化し，ゼータ変換を施すと $\mathrm{dp}[j] = g(j)$ になっている．Möbius 変換についても g と f が逆になること以外は同様である．ここで，j は部分集合を 2 進数でエンコードしたものである．

たとえば，Λ_n の各部分集合 S に対して式 (1) の包除原理を行うことを考える．すなわち，各 S に対して $f(S) = \sum_{T \subseteq S} (-1)^{|T|-1} \cdot \left|\bigcap_{i \in T} A_i\right|$ を計算する．これは，$g(\emptyset) = 0$，$g(T) = \left|\bigcap_{i \in T} A_i\right|$ として高速 Möbius 変換で得られる $f(S)$ を用いて $(-1)^{|S|} \cdot f(S)$ として計算できる．これにより，愚直には $O(3^{|S|})$ かかる計算を $O(|S| \cdot 2^{|S|})$ で行うことができる[*2]．

1.2.2 DP

式 (2) より，j 個の条件を満たす場合の数を求めることを考える．式 (4) を用いて j 個の条件に違反する場合の数を考えることもできる．

dp[i][j] では，i 番目までの条件を見て，そのうち j 個の条件を満たす場合の数（を求めるための値[*3]）として定義する．dp[i][j] から $\mathrm{dp}[i+1][j+1]$ への遷移（i 番目の条件を満たす）と $\mathrm{dp}[i+1][j]$ への遷移（i 番目の条件を考慮しない）をそれぞれがんばって考える．その後，$\mathrm{dp}[n][j]$ から $\sum_{|\Lambda|=j} \left|\bigcap_{i \in \Lambda} A_i\right|$ を求めることで式 (2) を計算できる．dp[n][j] が直接その値を表す場合，（遷移によっては）j を指数部の偶奇を示す {0, 1} のみにまとめることができうる．

また，式 (4) を用いる場合，$\mathrm{dp}[n][0]$ は「n 個目までの条件を見て 0 個の条件を考慮する場合」であり，これは集合全体に対応している．すなわち，以下のように簡単にできる（はず）．f は適当に定義する．

$$
\begin{aligned}
\left|\bigcap_{i=1}^{n} A_i\right| &= |A| - \sum_{j=1}^{n} (-1)^{j-1} \cdot \left(\sum_{|\Lambda|=j} \left| \bigcap_{i \in \Lambda} (A \setminus A_i) \right| \right) \\
&= f(0, \mathrm{dp}[n][0]) - \sum_{j=1}^{n} (-1)^{j-1} \cdot f(j, \mathrm{dp}[n][j]) \\
&= \sum_{j=0}^{n} (-1)^{j} \cdot f(j, \mathrm{dp}[n][j]).
\end{aligned}
$$

[*2] 目安としては，前者が $n \leqslant 15$ で後者が $n \leqslant 20$ 程度まで許容できる．
[*3] 遷移のしやすさと相談するとよさそう．

状態数が $O(n^2)$ になる場合，$n \leqslant 5000$ などでは MLE しうるので，一次元配列を使いまわすテクを使うとよい．また，操作回数など別のパラメータが必要になる場合は適宜 DP の次元を増やす必要がある（それはそう）．

1.2.3　約数系包除

ある n について $f(n)$ を求めた後，$g(n) = f(n) - \sum_{n' \sqsubset n} g(n')$ を求める[*4]局面が考えられる．典型的な例では，周期が関連する数え上げなどが挙げられる．より具体的には，たとえば周期 4 の数列には周期 2 や 1 の数列も含まれるので，最小周期が 4 の数列を数え上げたい場合にはそれらを取り除く必要がある．

さて，ここで以下の関係式が成り立つ（Möbius の反転公式）．

$$f(n) = \sum_{n' \sqsubseteq n} g(n') \iff g(n) = \sum_{n' \sqsubseteq n} \mu(n/n') \cdot f(n').$$

ここで，$\mu(n)$ は以下で定義される Möbius 関数である[*5]．

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1, \\ 0 & \text{if } {}^{\exists} m > 1 \text{ s.t. } m^2 \sqsubseteq n, \\ (-1)^k & \text{if } n = \prod_{i=1}^{k} p_i, \text{ where each } p_i \text{ is prime.} \end{cases}$$

$n' \sqsubseteq n$ である各 n' に対する $\mu(n')$ は，n の素因数の列挙をしておくと bit DP で求めることができる．また，n の上限によっては Eratosthenes の篩を応用して $1 \leqslant n' \leqslant n$ に対する $\mu(n')$ を計算しておくこともできる．

1.3　補足

$\mathcal{P}$ の各部分集合 $\mathcal{P}'$ について $\left|\bigcap_{i \in \mathcal{P}'} A_i\right|$ を前計算しておくことで，各クエリで与えられる $\mathcal{Q}$ について $\left|\bigcup_{i \in \mathcal{Q}} A_i\right|$ を高速に求められたりする．

$n \leqslant 10^9$，$n \leqslant 10^{18}$ なら n の持つ素因数の個数はそれぞれ，たかだか 9 個，15 個なので，n が多少大きくても n の素因数に関する部分集合であれば式 (1) をそのまま計算することができる．

[*4] $n' \sqsubseteq n$ は n' が n の約数であることを表し，$n' \sqsubset n$ は n' が n の真の約数である，つまり $n' \sqsubset n \iff n' \sqsubseteq n \land n' \neq n$ であることを表す．

[*5] $g(n)$ を $f(n')$ の足し引きで求める際に，足されるのか引かれるのか，あるいは打ち消されるのかを表す関数である．