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 $\mathcal{P} = \{P_1, \ldots, P_k\}$ を考える．$1 \leqslant i \leqslant k$ 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，$\bigcup_{i=1}^{k} A_i$ の要素数を求めるものである．$\Lambda_K = \{1, \ldots, k\}$ とする．

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

$|\Lambda| = |\Lambda'| \implies \left|\bigcap_{i\in\Lambda} A_i\right| = \left|\bigcap_{i\in\Lambda'} A_i\right|$ 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．

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

式 (1) においては $\sum$ で足し合わされる項が $2^k - 1$ 個だったのに対し，式 (2) では k 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 $\bigcap_{i=1}^{k} A_i$ の要素数を求めたいときには以下のようにするとよい．

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

すなわち，

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

これは，$A \setminus A_i$ を A_i と置き直すことで，式 (1) の枠組みで求められる．

1.1　証明

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

ここで，右辺の第 1 項と第 3 項を展開すると次のようになる．

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

[*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．

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

これらより，

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

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 dp[j] = f(j) で初期化し，ゼータ変換を施すと 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 bit でエンコードしたものである．

たとえば，Λ_k の各部分集合 S に対して $\sum_{T\subseteq S}(-1)^{|T|}\cdot\left|\bigcap_{i\in T}A_i\right|$ を考える．これは，$g(T)=\left|\bigcap_{i\in T}A_i\right|$ 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．

1.3 補足

1.3.1 暗黙の制約

$n\leqslant 10^9$，$n\leqslant 10^{18}$ 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．