アルゴリズムとデータ構造

その他の分割統治法

森 立平
mori@c.titech.ac.jp

2018 年 6 月 29 日

今日のメッセージ

- **分割統治法 = 「漸化式作ってそれをプログラムにするだけ」**
- **分割統治法の時間計算量のは漸化式を解けば得られる**
- ソート問題、選択問題以外にも分割統治法は使える

今日の目標

- 様々な分割統治法を習得する

1 その他の分割統治法

今まで分割統治法の代表例としてマージソートとクイックソートを考えてきたが、その他の分割統治法を紹介する。

2 Karatsuba 法

現在のコンピュータは基本的に 64 ビット整数の足し算や掛け算をする命令を持っている (64 ビット同士の掛け算は最大で 128 ビットになるが、その結果も得られる。ただし C 言語から直接はその機能は使えない)。巨大な数 (例えば数千、数万ビットの数を想像してみよう) の掛け算は 64 ビット同士の掛け算と足し算の組み合せで計算することができる。10 進数で一桁の掛け算、足し算しかできない我々が大きな数の掛け算ができるのと同様である。「一桁」の足し算と掛け算ができるコンピュータを使って n 桁の掛け算をするための時間計算量を考えよう。ここでいう時間計算量とは「一桁」の足し算と掛け算をする回数と定義する。この「一桁」というのは現在のコンピュータにとっては 0 以上 $2^{64}-1$ の値であるし人間にとっては 0 から 9 の値である。非負の n 桁の数 x と y について、その積 xy を計算するアルゴリズムを考えよう。通常の筆算を使うと時間計算量は $O(n^2)$ となる。しかし、分割統治法を使えばもっと高速に掛け算が計算できる。簡単のため、x と y の桁数 n は 2 の冪としよう (上位の桁に 0 を詰めればそのようにできるし桁数は高々 2 倍にしかならない)。x を上位 $n/2$ 桁 x_1 と下位 $n/2$ 桁 x_0 に分割する。一桁の数を 0 以上 $K-1$ 以下とし、$M=K^{n/2}$ とすると

$$x = x_1 M + x_0$$
$$y = y_1 M + y_0$$

が成り立つ。この表現を使えば

$$xy = x_1 y_1 M^2 + (x_1 y_0 + x_0 y_1) M + x_0 y_0$$

が得られる。この漸化式に従って得られる分割統治法のアルゴリズムの時間計算量 $\chi(n)$ は

$$\chi(n) = 4\chi(n/2) + cn$$

を満たす (足し算の時間計算量は $O(n)$ である)。この漸化式を解けば $\chi(n) = O(n^2)$ となり、筆算の時間計算量と同じオーダーになる。

一方で z_0, z_1, z_2 を次のように定義するとする。

$$\begin{aligned} z_0 &:= x_0 y_0 \\ z_1 &:= x_1 y_1 \\ z_2 &:= (x_0 + x_1)(y_0 + y_1) \end{aligned}$$

そうすると $z_2 - z_1 - z_0 = x_0 y_1 + x_1 y_0$ より

$$xy = z_1 M^2 + (z_2 - z_1 - z_0)M + z_0$$

が得られる。この漸化式に従って得られる分割統治法のアルゴリズムを Karatsuba 法と呼ぶ。この右辺では $n/2$ 桁の掛け算を 3 回しかしていないので、Karatsuba 法の時間計算量 $\chi(n)$ は

$$\chi(n) = 3\chi(n/2) + cn$$

を満たす。$b_n := \chi(n)/n$ とおくと、

$$b_n = \frac{3}{2} b_{n/2} + c$$

これを解くと

$$\begin{aligned} b_n + 2c &= \frac{3}{2}\left(b_{n/2} + 2c\right) \\ &= \left(\frac{3}{2}\right)^{\log n} (b_1 + 2c) \end{aligned}$$

よって $\chi(n) = O(3^{\log n}) = O(n^{\log 3})$ となる。$\log 3 \approx 1.585$ なので、通常の筆算よりも漸近的に高速であることが分かる。一方でもっと高速な整数乗算アルゴリズムも存在する。高速フーリエ変換を使うものには $O(n \log n \log\log n)$ 時間のアルゴリズムもある (しかし実際には n がかなり大きくないと Karatsuba 法など他のアルゴリズムよりも遅い)。

3 Strassen のアルゴリズム

$n \times n$ 行列同士の掛け算をするアルゴリズムを考えよう。行列の成分となっている数同士の掛け算と足し算の回数を時間計算量と定義する。$n \times n$ 行列 A と B について $C = AB$ とおくと

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

が成り立つ。この式に従って計算すると、1 成分あたり $O(n)$ 回の掛け算と足し算が必要なので、全体で $O(n^3)$ 時間かかる。

n を 2 冪と仮定する (行列に 0 成分を詰めることで 2 の冪に切り上げることができる)。$n \times n$ 行列 A, B, C を $(n/2) \times (n/2)$ 行列に分解して次の表現を得る。

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \qquad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

そうすると

$$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22} \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22} \end{aligned}$$

が成り立つ。この漸化式に従った分割統治法アルゴリズムの時間計算量 $\chi(n)$ は

$$\chi(n) = 8\chi(n/2) + cn^2$$

を満たす。これを解くと $\chi(n) = O(n^3)$ が得られる。一方で、行列 $P_1, \ldots, P_7$ を以下のように定義する。

$$\begin{aligned}
P_1 &:= A_{11}(B_{12} - B_{22}) \\
P_2 &:= (A_{11} + A_{12})B_{22} \\
P_3 &:= (A_{21} + A_{22})B_{11} \\
P_4 &:= A_{22}(B_{21} - B_{11}) \\
P_5 &:= (A_{11} + A_{22})(B_{11} + B_{22}) \\
P_6 &:= (A_{12} - A_{22})(B_{21} + B_{22}) \\
P_7 &:= (A_{11} - A_{21})(B_{11} + B_{12})
\end{aligned}$$

すると、

$$\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 \\
C_{12} &= P_1 + P_2 \\
C_{21} &= P_3 + P_4 \\
C_{22} &= P_5 + P_1 - P_3 - P_7
\end{aligned}$$

が成り立つ。$P_1, \ldots, P_7$ の計算には 7 回の $(n/2) \times (n/2)$ 行列の乗算が必要なので、この漸化式に従った分割統治法アルゴリズムの時間計算量 $\chi(n)$ は

$$\chi(n) = 7\chi(n/2) + cn^2$$

を満たす。この漸化式を解くと $\chi(n) = O(n^{\log 7})$ が得られる。$\log 7 \approx 2.807$ なので、通常のアルゴリズムよりも漸近的に高速であることが分かる。一方でもっと高速な行列乗算アルゴリズムも存在する。2018 年 6 月現在で最も高速な行列乗算アルゴリズムの時間計算量は $O(n^{2.3728639})$ である。

4 高速ゼータ変換

集合 $\{1, 2, \ldots, n\} =: [n]$ 上の集合関数 $f\colon 2^{[n]} \to \mathbb{R}$ についてゼータ変換を次のように定義する。

定義 1. ゼータ変換 $\widehat{f}\colon 2^{[n]} \to \mathbb{R}$ を

$$\widehat{f}(T) := \sum_{S \subseteq T} f(S)$$

と定義する。

このゼータ変換から f に戻すことができる。

定理 1.

$$f(S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} \widehat{f}(T)$$

Proof.

$$\begin{aligned}
\sum_{T \subseteq S} (-1)^{|S|-|T|} \widehat{f}(T) &= \sum_{T \subseteq S} (-1)^{|S|-|T|} \sum_{V \subseteq T} f(V) \\
&= \sum_{V \subseteq S} \left(\sum_{V \subseteq T \subseteq S} (-1)^{|S|-|T|} \right) f(V).
\end{aligned}$$

ここで

$$\sum_{V\subseteq T\subseteq S}(-1)^{|S|-|T|}=\begin{cases}1, & \text{if } V=S\\ 0, & \text{otherwise.}\end{cases}$$

となることを示せばよい。$V=S$ のときに左辺が 1 となるのは明らか。$V\subsetneq S$ のとき任意に選んだ $x\in S\setminus V$ に対して

$$\sum_{V\subseteq T\subseteq S}(-1)^{|S|-|T|}=\sum_{V\subseteq T\subseteq S\setminus\{x\}}\left((-1)^{|S|-|T|}+(-1)^{|S|-|T\cup\{x\}|}\right)=0.$$

□

ゼータ変換の時間計算量を考えよう。整数の四則演算は定数時間でできると仮定しよう。ゼータ変換を定義に従って素朴に計算すると、すべての要素を計算するのに $O(4^n)$ 時間かかる(少し工夫すると $O(3^n)$ にはなる)。高速ゼータ変換と呼ばれるアルゴリズムはゼータ変換を $O(2^n n)$ 時間で計算する。集合関数 $f\colon 2^{[n]}\to\mathbb{R}$ に対して、$f_0,f_1\colon 2^{[n-1]}\to\mathbb{R}$ を

$$f_0(S):=f(S),\qquad f_1(S):=f(S\cup\{n\}),\qquad S\subseteq[n-1]$$

と定義すると、

$$\widehat{f}(T)=\begin{cases}\widehat{f_0}(T), & \text{if } n\notin T\\ \widehat{f_0}(T\setminus\{n\})+\widehat{f_1}(T\setminus\{n\}), & \text{otherwise}\end{cases}$$

という漸化式が得られる。この漸化式に従ってゼータ変換を計算するのが高速ゼータ変換である。

$$\chi(n)=2\chi(n-1)+c2^n$$

より $\chi(n)=O(2^n n)$ が得られる。高速ゼータ変換の C 言語プログラムは次のようになる。

```
void zeta(int A[], int n){
  int i;
  int m = 1 << n;
  if(n == 0) return;
  zeta(A, n - 1);
  zeta(A + m / 2, n - 1);
  for(i = 0; i < m / 2; i++){
    A[i + m / 2] += A[i];
  }
}
```

これを展開して、再帰ではなく反復でゼータ変換を書くこともできる。詳しくは説明しない。

```
void zeta(int A[], int n){
  int i, j;
  int m = 1 << n;
  for(i = 1; i < m; i <<= 1){
    for(j = 0; j < m; j++){
      if(j & i) A[j] += A[j ^ i];
    }
  }
}
```

この高速ゼータ変換を用いれば、色々な問題が高速に解けるようになる。

問題 1. 集合 $[n]:=\{1,2,\dots,n\}$ の部分集合 $S_1,S_2,\dots,S_m$ と整数 $k\in[m]$ が与えられているとき、組 $(i_1,\dots,i_k)\in[m]^k$ で $\bigcup_{j=1}^{k}S_{i_j}=[n]$ となるものの個数を求めよ。

解き方. 素朴に全探索すると m^k 時間はかかってしまうが、この問題には $O(2^n n)$ 時間アルゴリズムが存在する。解に対応する集合関数 $g\colon 2^{[n]} \to \mathbb{R}$ を

$$g(S) := \left| \left\{ (i_1, \ldots, i_k) \in [m]^k \mid \bigcup_{j=1}^{k} S_{i_j} = S \right\} \right|$$

と定義する。すると $g([n])$ が解となる。g のゼータ変換 $\widehat{g}$ は

$$\widehat{g}(T) = \sum_{S \subseteq T} g(S) = \left| \left\{ (i_1, \ldots, i_k) \in [m]^k \mid \bigcup_{j=1}^{k} S_{i_j} \subseteq T \right\} \right|$$

を満たす。一方で

$$f(S) := |\{i \in [m] \mid S_i = S\}|$$

と定義すると

$$\widehat{f}(T) = \sum_{S \subseteq T} f(S) = |\{i \in [m] \mid S_i \subseteq T\}|$$

を満たす。よって $\widehat{g}(S) = \widehat{f}(S)^k$ である。

よって f から高速ゼータ変換で $\widehat{f}$ を計算し、各成分を k 乗することで $\widehat{g}$ が求まる。そして定理 1より $\widehat{g}$ から $g([n])$ が高々 $O(2^n n)$ 時間で計算できる。よって全体で $O(2^n n)$ 時間で計算ができた。 □

問題 2. 集合 $[n] := \{1, 2, \ldots, n\}$ の部分集合 $S_1, S_2, \ldots, S_m$ が与えられているとき、集合 $K \subseteq [n]$ で $\bigcup_{i \in K} S_i = [n]$ を満たすものの最小のサイズ $|K|$ を求めよ。

この問題は問題 1と二分探索を使えば $O(2^n n \log n)$ 時間で解ける。

問題 3. 集合 $[n] := \{1, 2, \ldots, n\}$ の部分集合 $S_1, S_2, \ldots, S_m$ と整数 $k \in [m]$ が与えられているとき、集合 $\{i_1, \ldots, i_k\} \subseteq [m]$ で $\bigcup_{j=1}^{k} S_{i_j} = [n]$ となるものの個数を求めよ。

解き方. 解に対応する集合関数 $g\colon 2^{[n]} \to \mathbb{R}$ を

$$g(S) := \left| \left\{ \{i_1, \ldots, i_k\} \subseteq [m] \mid \bigcup_{j=1}^{k} S_{i_j} = S \right\} \right|$$

とおくと $g([n])$ が解となる。g のゼータ変換 $\widehat{g}$ は

$$\widehat{g}(T) = \sum_{S \subseteq T} g(S) = \left| \left\{ \{i_1, \ldots, i_k\} \subseteq [m] \mid \bigcup_{j=1}^{k} S_{i_j} \subseteq T \right\} \right|$$

を満たす。一方で

$$\widehat{f}(T) = \sum_{S \subseteq T} f(S) = |\{i \in [m] \mid S_i \subseteq T\}|$$

よって

$$\widehat{g}(T) = \binom{\widehat{f}(T)}{k}$$

である。高速ゼータ変換を用いれば、問題 1と同様に $O(2^n n)$ 時間で問題 4が解ける. □

問題 4. 集合 $[n] := \{1, 2, \ldots, n\}$ の部分集合 $S_1, S_2, \ldots, S_m$ が与えられているとき、集合 $V \subseteq [m]$ で $\bigcup_{i \in V} S_i = [n]$ となるものの個数を求めよ。

解き方. 解に対応する集合関数 $g\colon 2^{[n]} \to \mathbb{R}$ を

$$g(S) := \left| \left\{ V \subseteq [m] \mid \bigcup_{i \in V} S_i = S \right\} \right|$$

とおくと $g([n])$ が解となる。g のゼータ変換 $\widehat{g}$ は

$$\widehat{g}(T) = \sum_{S \subseteq T} g(S) = \left| \left\{ V \subseteq [m] \mid \bigcup_{i \in V} S_i \subseteq T \right\} \right|$$

を満たす。一方で

$$\widehat{f}(T) = \sum_{S \subseteq T} f(S) = |\{ i \in [m] \mid S_i \subseteq T \}|$$

よって

$$\widehat{g}(T) = 2^{\widehat{f}(T)}$$

である。高速ゼータ変換を用いれば、問題 1と同様に $O(2^n(n+m))$ 時間で問題 4が解ける． □