グラフとアルゴリズムとプログラムのやさしいおはなし

渡邉敏正

2021 年 5 月 13 日

第 2 回

1　総当たり法 (力ずくの方法)

答を得るために可能なすべての場合を調べる**力ずくの方法**（brute-force method）について 1 つ話題を提供します。力ずくの方法を**総当たり法**とよぶこともあります。以下では総当たり法の名称を使います。

1.1　整数の和を考えてみる

まず、1 つの例を考えてみましょう。

4 つの整数 $1, 3, 5, 8$ ともう 1 つの整数 14 があるとき、4 つの数値からいくつか選んで総和を作るとそれがちょうど 14 になるかどうか、イエスかノーかという問題について考えてみましょう。さて、皆さんはどのようにしてイエスかノーかを決めるでしょうか。

うまく $1, 5, 8$ を選んだ人はすぐにイエスと答えるでしょう。あるいは、いくつかの数値の和を試してみてこの組合せに行き着いてやはりイエスという方もいるでしょう。あとの説明のために、この問題を次のように少し言い直しておきます：

> 「正の整数の集合 $S = \{1, 3, 5, 8\}$ と別の正の整数 $b = 14$ があるとき、S の中から何個か選んでその総和がちょうど $b = 14$ に等しくなることはあるか否か」

という問題です。さらに、「S から何個か選んで」というところをもう少し変えてみると、

> 「正の整数の集合 $S = \{1, 3, 5, 8\}$ と別の正の整数 $b = 14$ があるとき、S は部分集合 S' として、S' に含まれる数値の総和がちょうど $b = 14$ に等しいようなものを持つか否か」

という問題になります。S の中に要素の総和が b であるような部分集合 S' があるかどうかを問う問題です。

では、同じ集合 $S = \{1, 3, 5, 8\}$ ですが今度は総和のターゲット b が少し変わって $b = 15$ の場合はどうでしょうか。

ここで、恐らく多くの方が実行するであろうと予想される総当たり法をやってみましょう。表 1 を見てください。$S = \{1, 3, 5, 8\}$ すべての部分集合について、その要素の総和を作って $b = 15$ に等しいかどうかチェックする方法です。表 1 は私がやってみたチェックの様子を示しています。恐らくみなさんが実行した総当たりのチェック方法もこれと大差はないと思います。$\emptyset$ は空集合（要素が何もない集合）を表しています。

総当たりですべての可能な場合をチェックしながらこのような表を作ってみれば、答はノーであることがわ

表 1　総当たり法の実行過程の例

No.	部分集合	和
0	∅	0
1	{1}	1
2	{3}	3
3	{5}	5
4	{8}	8
5	{1,3}	4
6	{1,5}	6
7	{1,8}	9
8	{3,5}	8
9	{3,8}	11
10	{5,8}	13
11	{1,3,5}	9
12	{1,3,8}	12
13	{1,5,8}	14
14	{3,5,8}	16
15	{1,3,5,8}	17

かります。

ついでに言いますと、$b = 14$ の場合で表 1 の 1 番目から探したとすると、14 回目でイエスがわかったはずです。全部を見た場合に近い回数ですね。では、全体集合 $S = \{1, 3, 5, 8\}$ から始めて要素を減らしていく方法で、たとえば表 1 の逆順で探した方はどうでしょう。すぐに 3 回目でイエスがわかったことでしょう。こちらがずっと早く答がわかった訳ですが、これはやってみてわかったことであって、探し始める前には予想がつかないことです。

1.2　部分集合の個数

元の話題に戻りますが、「なんだ、こんな簡単なことのどこに問題があるのだろう？」と疑問に思われる方がいるかもしれませんので、もう少し説明を加えていきます。着目点は総当たりの場合にチェックした回数です。

ターゲットが $b = 14$ の場合には運がよければすぐに答がわかりました。でも、$b = 15$ のときはすべての場合を見ることになってしまいました。問題の中での値の設定によって答を得るまでに行うチェックの回数が違います。どのような回数に着目すればいいでしょうか。いろいろな考え方があると思いますが、ここでは最も多くかかる場合を見てみましょう。最も多い場合でどれくらいチェックすれば答が得られるか、という視点で見てみようという訳です。このような回数の見積りの仕方を「最悪の場合の見積り」などといいます。ただし、この回数が多いということは、いつもチェックの回数が多いということを意味するのではない、ということは覚えておいてください。では、この問題について最悪の場合にどれくらいのチェック回数になるかを考えてみましょう。

表 1 を見ていただくと分かると思いますが、答がノーである場合にすべての可能な場合をチェックすることになり、これが最悪の回数です。答がイエスかノーかは予め知ることはできない、前もっては誰にもわからない、ということに注意してください。すべての場合のチェックは結局は S のすべての部分集合について総和を見ていることになります。したがってより正確に言えば、最悪の場合の着目点は**すべての部分集合の個数**ということになります。集合 S の要素数を $|S|$ と表すことにします。

上記の例では $|S| = 4$ ですが、部分集合は何個あるでしょうか。表から分かるように全部で 16 個ですが、この個数はどういう数値でしょうか。実は、$2^{|S|} = 2^4 = 16$ という値です。なぜこのような値になるのでしょ

うか。皆さんで考えてみてください。この例だけでなく、集合 S の要素数が $n = |S|$ のとき、S のすべての部分集合の個数は必ず 2^n になります。すべての部分集合には空集合も全体集合（S 自身）も含まれます。

1.3 完了までの時間

では、集合 S の要素数が $n = |S|$ のとき、n と部分集合の個数との関係をいくつか実際に求めてみましょう。なお、数値を簡単にするため

$$2^{10} = 1024 \approx 1000 = 10^3$$

と近似して個数を見積もってみます。$\approx$ は近似の意味ですが、この近似では個数を少なめに見積ることになります。ただ、個数ではイメージが分かりにくいと思いますので、

> 『S の 1 つの部分集合を選び，要素総和を求めて b に等しいか否かを判定する』
> という操作が 1 秒間に百万回実行できるような高速なコンピュータがあると仮定し、

この仮定の下で、要素数 n の集合の部分集合をすべてチェックした場合にかかる時間を見てみることにします。この時間を T_n と表すことにします。たとえば、$|S| = 10$ ならば部分集合は 2^{10} 個ですから、これを 1 秒あたりにチェックできる個数、百万 $= 10^6$ で割れば全部のチェックにかかる秒数 T_{10} がわかります。いくつかを示すと以下のようになります。

$n = 10$ のとき：$T_{10} = 2^{10} \times 10^{-6} \approx 10^3 \times 10^{-6} = 10^{-3}$（秒）
$n = 30$ のとき：$T_{30} = 2^{30} \times 10^{-6} \approx 10^9 \times 10^{-6} = 10^3$（秒）（= 16 分 40 秒）
$n = 50$ のとき：$T_{50} = 2^{50} \times 10^{-6} \approx 10^{15} \times 10^{-6} = 10^9$（秒）（> 25 年）
$n = 100$ のとき：$T_{100} = 2^{100} \times 10^{-6} \approx 10^{30} \times 10^{-6} = 10^{24}$（秒）（$> 2.5 \times 10^{16}$年）

ここでは 1 日や 1 年の秒数を以下のように非常に大まかに見積もっています。

$$\begin{aligned}
&24\text{ 時間（1 日）} &&= 86,400\text{ 秒} < 10^5\text{秒},\\
&365\text{ 日（1 年）} &&= 31,536,000\text{ 秒} < 4 \times 10^7\text{秒},\\
&T_{100} &&\approx 10^{24}\text{(秒)} = 10^{15}\text{(秒)} \times 10^9\\
& &&> 2.5 \times 10^{16}\text{(年)}\ (\text{1 年を } 4 \times 10^7\text{秒と多めに見積って})\\
& &&> 4.6 \times 10^9\text{(年)} \times 540\text{ 万 (回)}\ (\text{地球の年齢との比較})
\end{aligned}$$

T_{50} の 25 年以上（1 年を 4×10^7秒 で見積り) というのも長いですね。さらに見ていくと、地球ができて 46 億年（$= 4.6 \times 10^9$ 年）程度ですが，T_{100} はその 540 万倍よりも長い我々の想像をはるかに超える時間です．それは確かに有限の時間ですが，私達には測定不可能な時間のようです。太陽が中心核で燃料となる水素を使い果たして「恒星の死」に向かって赤色巨星へと変化していき、今から 60〜80 億年後の地球は巨大化した太陽に飲み込まれていないかもしれないが灼熱地獄だろう、と言われています。ですから 46 億年の 2 回目さえ考え難いのです。(あくまで、地球という星の上での現在の状況から判断して、ということです。人類が別の星に移住したり、あるいはこの問題の解を瞬時に得る方法やシステムなどを獲得すれば状況は変わるかもしれません。)

総当たり法によって答を探していると、最悪の場合には、待てど暮らせど終わらないような単純な問題が身近にあることがわかっていただけたと思います。このような問いかけを**解きにくい問題** (intractable problem) とよぶ場合もあります。

今回紹介した問題は**部分和問題** (subset-sum problem) という名前が付いています。現在までの長い間の研究にもかかわらず、残念ながら、最悪の時間が総当たり法より短い解法は見つかっていません。ですから、この問題については現状では総当たり法に頼ることになります。だからといってどの問題についても、(最悪の場合に) 長時間かかってしまう総当たり法に飛びついてしまうのではなく、まず、もっと手早く答に辿り着く方法 − 効率的な方法 − を探そうとすることは自然なことだと思います。

注 2.1　部分和問題を比較的効率良く解く方法として動的計画法 (dynamic programming) があります。これについては別途説明します。