集合の分割の列挙

●問題

入力として非負整数 n が与えられたときに，集合 $\{1,\dots,n\}$ を非空集合の族へと分割する方法すべてを，重複なく列挙したい．以下の設問に答えよ．

(1) 集合の分割を列挙するプログラムを作成せよ．
(2) プログラムの動作を解説せよ．
(3) 利用したデータ構造やアルゴリズムについて論じよ．
(4) 実行時間について実験・考察し，$n = 15$ の場合の実行時間を推測せよ．
(5) 可能な別解について論じよ．

●解答

(1) プログラム (記述言語は Python) を以下に示す．

```
1: def partitions(seq):
2:     """列表現された集合の分割を再帰的に列挙"""
3:     if len(seq) == 0: return [[]]
4:     front, last = seq[:-1], seq[-1:]
5:     return [new_seq for p in partitions(front) for new_seq in expand(p, last)]
6:
7: def expand(p, last):
8:     """要素を一つ加えて分割を拡張"""
9:     return [p[:i] + [p[i]+last] + p[i+1:] for i in range(len(p))] + [p+[last]]
```

(2) 上記のプログラムを解説する．関数 `partitions` は，集合 $S = \{1,\dots,n\}$ を表す列 `seq` を受け取り，S の可能な分割すべてを列にして返す．$n = 0$ つまり $S = \emptyset$ のとき，S は 0 個の集合族の和としてのみ表せるので $\emptyset$ だけを返す (3 行目)．$n \geq 1$ のときは，$S = \{1,\dots,n-1\} \cup \{n\}$ のように前半 `front` と最後 `last` に分けて (4 行目)，1 要素減った集合 $\{1,\dots,n-1\}$ に対して再帰的に分割すべてを求めてから，各分割に要素 n を追加する方法をすべて列挙する (5 行目)．分割への要素の追加には関数 `expand` を使う．すでに求めた分割 $\mathcal{P} = \{P_1,\dots,P_k\}$ に要素 n を追加する方法には 2 通りある．分割を構成する集合のどれか P_i に要素 n を加える方法 (9 行目前半) と，分割 $\mathcal{P}$ に n 単独の集合 $\{n\}$ を新たに加える方法 (9 行目後半) である．

例えば $n = 2$ の場合の $\{1,2\}$ の分割は $\{\{1,2\}\}$ と $\{\{1\},\{2\}\}$ の 2 通りで，`partitions([1,2])` は

```
[ [[1,2]], [[1],[2]] ]
```

を返す．それぞれの分割に対して要素 3 を加える．`expand([[1,2]], 3)` は

```
[ [[1,2,3]], [[1,2],[3]] ]
```

を返し，`expand([[1],[2]], 3)` は

```
[ [[1,3],[2]], [[1],[2,3]], [[1],[2],[3]] ]
```

を返すので，これら 5 個の分割を並べた

```
[ [[1,2,3]], [[1,2],[3]], [[1,3],[2]], [[1],[2,3]], [[1],[2],[3]] ]
```

が `partitions([1,2,3])` の実行結果である．これは，$\{1,2,3\}$ の5通りの分割 $\{\{1,2,3\}\}$, $\{\{1,2\},\{3\}\}$, $\{\{1,3\},\{2\}\}$, $\{\{1\},\{2,3\}\}$, $\{\{1\},\{2\},\{3\}\}$ すべてを列で表している．

(3) 利用したデータ構造やアルゴリズムについて述べる．集合の分割を求める上記のアルゴリズムでは，集合に常に新しい要素を加えるため，要素の重複の有無を調べる必要がなく，集合データ型を使わずに処理できる．リストは Python の組み込みデータ型であり，走査や分解と連結を簡潔に書けるため，集合の分割を表すデータ構造としてリストのリストを採用した．アルゴリズムには，より小さな集合に対する解を使って必要な解を構成する分割統治法を使い，分割の列挙を再帰関数として実現した．

(4) 実行時間について実験・考察する．作成したプログラムで入力 n を変えて，実行にかかる時間を計測した結果を表 1 に示す．

表 1: 分割の列挙の実行時間

n	実行時間 (秒)
0	0.000033
1	0.000023
2	0.000024
3	0.000067
4	0.000119
5	0.000386
6	0.001498
7	0.022714
8	0.269919
9	1.527642
10	9.381752
11	61.331102
12	420.344510

実行時間への影響が大きいのは，分割の総数である．n 元集合を k 分割する方法の総数を S_k^n で表すと，S_k^n は次の漸化式を満たす．

$$\mathrm{S}_k^n = \begin{cases} 1 & (0 \le k = n) \\ k \cdot \mathrm{S}_k^{n-1} + \mathrm{S}_{k-1}^{n-1} & (1 \le k < n) \\ 0 & (\text{その他}) \end{cases}$$

第 1 の場合には，n 要素を n 分割するので，すべてが単独の要素からなる $\{\{1\},\{2\},\ldots,\{n\}\}$ というただ一つの分割方法しかない．第 2 の場合には，(2) で述べたアルゴリズムに沿って分割の総数を考える．第 1 項 $k \cdot \mathrm{S}_k^{n-1}$ は，集合 $\{1,\ldots,n-1\}$ の各 k 分割に対して，分割を構成する k 個の集合から一つを選んで要素 n を追加する方法の総数を表す．第 2 項 S_{k-1}^{n-1} は，集合 $\{1,\ldots,n-1\}$ の各 $k-1$ 分割に対して要素単独の集合 $\{n\}$ を追加して，k 分割にする方法の総数を表す．したがって，n 元集合の分割の総数 B_n は次式で表せる．

$$\mathrm{B}_n = \sum_{k=0}^{n} \mathrm{S}_k^n$$

組み合わせ論では，ここで使った S_k^n を第 2 種スターリング数と呼び，B_n をベル数と呼ぶ．n が 0 から 15 の範囲のベル数を表 2 に示す．

表 2: ベル数

n	B_n	n	B_n
0	1	8	4140
1	1	9	21147
2	2	10	115975
3	5	11	678570
4	15	12	4213597
5	52	13	27644437
6	203	14	190899322
7	877	15	1382958545

n 元集合の分割一つの出力時間が n に比例すると (大まかに) 見積もれば，分割すべてを列挙するための推定実行時間 t_n は $c \cdot n \cdot \mathrm{B}_n$ で表せる．$n = 12$ のときの測定値を使って，係数 c の値を求める．

$$c = \frac{t_n}{n \cdot \mathrm{B}_n} = \frac{420.344510}{12 \times 4213597} \approx 8.31 \times 10^{-6}$$

これを基に，実行時間の実測値と推定値を片対数グラフで表示したものを図 1 に示す．$n \geq 8$ の場合に適切な推定値が得られていることを確認できる．

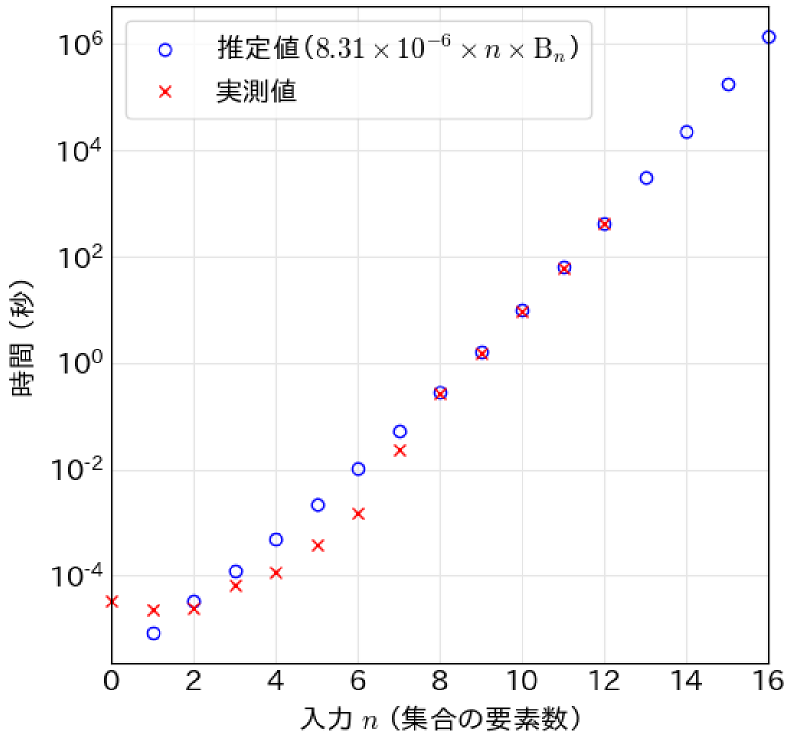

図 1: 実行時間の実測値と推定値

以上の結果から，$n = 15$ の場合の実行時間を推定する．

$$t_n = c \cdot n \cdot \mathrm{B}_n \approx (8.31 \times 10^{-6}) \times 15 \times 1382958545 \approx 1.72 \times 10^5 (\text{秒})$$

つまり，プログラムと実行環境が同じなら，$n = 15$ の場合の実行に 48 時間程度必要だと推定できる．

(5) 別解について考察する．データ構造としては，分割を (要素の列の列などで) 直接表す他に，各要素が分割を構成する何番目の集合に属すかを (列や写像で) 表す方法もある．例えば，集合 $\{1,2,3,4\}$ の分割 $\{\{1\},\{2,4\},\{3\}\}$ は，要素 1, 2, 3, 4 が分割を構成する集合 $\{1\}$, $\{2,4\}$, $\{3\}$, $\{2,4\}$ にそれぞれ属すので，列 $[1,2,3,2]$ で表せる．記憶量が少なくて済むが，走査しただけでは集合族を出力できず，出力時に集合族を構成する手間がかかる．アルゴリズムとしては，分割すべてを再帰的に構成する以外にも，分割を一つ受け取って (あるいは記憶して) 次の分割を構成することを繰り返しても，分割を列挙できる．記憶量が最小限に抑えられ，列挙の中断や再開が自由にできるが，再帰と比べてアルゴリズムが複雑になる．

●参考文献

集合の分割の列挙などの組み合わせアルゴリズムについては，次の文献が参考になる．

- “Combinatorial Algorithms”, A. Nijenhuis and H. S. Wilf, second edition, Academic Press, 1978. `https://www.math.upenn.edu/~wilf/website/CombAlgDownld.html`

集合の分割の総数 (第 2 種スターリング数とベル数) については，次の二つの文献が参考になる．

- 『コンピュータの数学』，R. L. グレアム，D. クヌース，O. パタシュニク，共立出版，1993.
- 『数え上げ組合せ論入門』，成嶋弘，日本評論社，2003.

山田 俊行

`https://www.cs.info.mie-u.ac.jp/~toshi/`