

プログラミング創造演習・2023 年度問題

解答期限: 2023 年 4 月 12 日 (水) 17:00

1 順列の列挙

コード 1 は整数 0, 1, 2 の順列を列挙する C のプログラムであり, 実行すると標準出力に右のように出力する.

```
0, 1, 2
0, 2, 1
1, 0, 2
1, 2, 0
2, 0, 1
2, 1, 0
```

問 1.1. コード 1 の A に入る式を答えよ. 新たな関数は追加しないこと.

問 1.2. コード 1 を実行したときの関数 bt の呼び出し回数を答えよ. 関数 main による最初の呼び出しも数えるものとする.

問 1.3. コード 1 を実行したときの関数 ck の呼び出し回数を答えよ.

問 1.4. コード 1 の 8 行目を `bt(a, n, 0);` として実行することで $0, 1, \dots, n-1$ の順列を列挙できる. $n = 10$ として実行したところ, 出力に 5, 4, 6, 8, 7, 3, 1, 0, 9, 2 という行があった. 何行目か.

問 1.5. n を非負整数とし, コード 1 の 8 行目を `bt(a, n, 0);` として実行したときの関数 bt の呼び出し回数を T_n とする. 以下の漸化式がなりたつよう, B および C に入る式を答えよ (コード 1 では配列 a の長さは 256 となっているが, この問題に関しては a は十分長いと考えてよい).

$$T_0 = 1, \quad T_{n+1} = (\text{B})T_n + \text{C}$$

コード 1 順列を列挙するプログラム (前半)

```
1 #include <stdio.h>
2
3 void bt(int *, int, int);
4
5 int a[256];
6
7 int main(void) {
8     bt(a, 3, 0);
9     return 0;
10 }
11
12 void printa(int a[], int n) {
13     if (n > 0) {
14         printf("%d", a[0]);
15         for (int i = 1; i < n; i++) {
16             printf(", %d", a[i]);
17         }
18         printf("\n");
19     }
20 }
21
```

コード 1 順列を列挙するプログラム (後半)

```
22 int ck(int a[], int x, int y) {
23     for (int k = 0; k < x; k++) {
24         if (A)
25             return 0;
26     }
27     return 1;
28 }
29
30 void bt(int a[], int n, int x) {
31     if (x == n) {
32         printa(a, n);
33     }
34     else {
35         for (int y = 0; y < n; y++) {
36             if (ck(a, x, y)) {
37                 a[x] = y;
38                 bt(a, n, x + 1);
39             }
40         }
41     }
42 }
```

2 n-クイーン問題

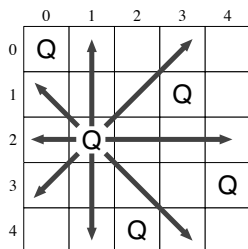


図1 クイーンの配置

n -クイーン問題とは、大きさ $n \times n$ のチェス盤上に n 個のクイーンを互いの効き筋上にないように配置する全ての方法を列挙する問題である。ここでは対称解同士は別のものとみなす。チェスのクイーンの効き筋は、縦、横および斜め 45° 方向である。図1は $n = 5$ の場合の解のひとつである。図中の矢印は座標 $(1, 2)$ に配置したクイーンの効き筋を表している。

ここでクイーンの配置を長さ n の整数列 q_0, q_1, \dots, q_{n-1} で表すことにする。各 q_i は $0 \leq q_i < n$ をみたす整数である。この数列は $0 \leq i < n$ をみたす各 i について座標 (i, q_i) にクイーンが配置されていることを表している。例えば図1は数列 $0, 2, 4, 1, 3$ として表される。

問2.1. 実はコード1の **A** に適切な式を与え、8行目を `bt(a, n, 0);` とすることで n -クイーン問題の解を上で説明した数列の形で列挙させることができる。例えば $n = 5$ の場合は右のような出力（一部省略）になる。 **A** に入る式を答えよ。新たな関数は追加しないこと。

```
0, 2, 4, 1, 3
0, 3, 1, 4, 2
1, 3, 0, 2, 4
...
4, 2, 0, 3, 1
```

コード2 再帰を使わないbtの定義

```
30 void bt(int a[], int n, int x) {
31     int y = 0;
32     while (1) {
33         while (y < n) {
34             if (ck(a, x, y)) {
35                 a[x] = y;
36                 x = D;
37                 if (x == n) {
38                     printa(a, n);
39                     E
40                 }
41                 y = 0;
42             }
43             else {
44                 y = F;
45             }
46         }
47         x = x - 1;
48         if (G) return;
49         H
50     }
51 }
```

問2.2. コード1の **A** を問2.1のように変更し、 $n = 8$ として実行したときの関数 `bt` の呼び出し回数を答えよ。

問2.3. コード1の **A** を問2.1のように変更し、 $n = 8$ として実行したときの関数 `ck` の呼び出し回数を答えよ。

問2.4. コード1の関数 `bt` では再帰呼び出しを使っているが、同じ関数を再帰を用いないように実現したものがコード2である。 **D**, **F**, **G** に入る式および **E**, **H** に入る文を答えよ。新たな関数は追加せず、出力は解の順番を含めコード1のものと同一になるようにすること。また、本問題を解くにあってどのように考えたかを簡単に説明せよ。

問2.5. コード1の **A** を問2.1のように変更し、関数 `bt` をコード2に置き換えたものを $n = 8$ として実行したときの関数 `ck` の呼び出し回数を答えよ。

問2.6. 大きさ 8×8 のチェス盤上の相異なる座標に $8 \times k$ 個のクイーンを配置する。ただし、盤面上のクイーンを8個ずつ k 個のグループに分けたとき、各グループが8-クイーン問題の解となるようにしたい。このような条件をみたす正の整数 k の最大値を答えよ。また、本問題をどのようにして解いたかを簡単に説明せよ。