

情報 練習問題 1

【問題】

以下の文章を読んで、問いに答えよ。

ハノイの塔とは、以下のルールに従って、すべての円盤を右側の杭に移動する問題である。

- 3本の杭と、中央に穴の開いた大きさの異なる複数の円盤から構成される。
- 最初はすべての円盤が左側の杭に小さいものが上になるように順に積み重ねられている。
- 円盤を一回に一枚ずつどれかの杭に移動させることができるが、小さな円盤の上に大きな円盤を乗せることはできない。

この問題を解くために、以下のような関数 f を考えた。

```
f(n, a, b) =  
  if (n > 1) then f(n - 1, a, 6 - a - b) endif  
  “(a の値) (b の値)” と画面に表示して改行  
  if (n > 1) then f(n - 1, 6 - a - b, b) endif
```

ただし、はじめに円盤が積まれている杭の番号を 1 (左側の杭)、移動させたい杭の番号を 3 (右側の杭)、のこりを 2 (中央の杭) とする。

この関数によって、円盤が右側の杭に n 枚積まれているとき、 $f(n, 1, 3)$ を呼び出すことによって、画面にすべての操作が表示される。

- (1) 引数 n, a, b がそれぞれ何を示しているか述べよ。
- (2) $f(3, 1, 3)$ を実行した際に、画面に表示される内容を示せ。
- (3) f 中の $6 - a - b$ の意味を述べよ。
- (4) $f(n, 1, 3)$ を呼び出した際の、操作回数 (円盤を移動する回数) を求めよ。ただし、考え方も述べること。
- (5) f を非再帰的に実現することを考える。どのような方法を用いて再帰を除去すればよいか述べよ。(発展問題)

(解答は次ページです)

【解答】

(1)

 n : 移動する円盤の枚数 a : 移動元の軸番号 b : 移動先の軸番号

(2)

 $1 \rightarrow 3$ $1 \rightarrow 2$ $3 \rightarrow 2$ $1 \rightarrow 3$ $2 \rightarrow 1$ $2 \rightarrow 3$ $1 \rightarrow 3$

(3)

中間軸（移動先・移動元でもない軸）の番号である．

(4)

$f(n, 1, 3)$ における移動回数を a_n とすると、(どこの軸が移動元・移動先であろうと、移動回数には関係しない)

 f の定義から、

$$a_{n+1} = a_n + 1 + a_n = 2a_n + 1$$

 $a_1 = 1$ より、 $a_n = 2^n - 1$ ゆえに $2^n - 1$ 回．

(5)

スタックを利用する．

具体的には、以下の通り．

- (n, a, b) の値をスタックにプッシュする．
- $(n, a, b) \rightarrow (n-1, a, 6-a-b)$ に書き換え、関数のはじめにもどる．
- 移動を表示する．
- スタックから (n, a, b) の値をポップし、移動を表示する．
- $(n, a, b) \rightarrow (n-1, 6-a-b, b)$ に書き換え、関数のはじめにもどる．

参考のため、C によるソースコードを示す．スタックは配列で実現してある．
C++ を使うならば、STL の `stack`（もしくはより上位の `deque`）でもよい． f は `move`、`goto` を気持ち悪く感じる人は `while(true)` と `continue` で．

```
#include <stdio.h>
```

```
typedef struct {
    int no; // 移動する円盤
    int pos; // 出発位置
    int dest; // 移動位置
} Hanoi;
```

```
void move(int no, int x, int y)
{
    Hanoi stk[100];
```

```
    int ptr = 0;
Top:
    if (no > 1) {
        stk[ptr].no = no;
        stk[ptr].pos = x;
        stk[ptr].dest = y;
        ptr++;

        no = no - 1;
        y = 6 - x - y;
        goto Top;
    }

    printf("%d  %d", x, y);

    if (ptr > 0) {
        ptr--;
        no = stk[ptr].no;
        x = stk[ptr].pos;
        y = stk[ptr].dest;
        printf("%d  %d", x, y);

        if (no > 1) {
            no = no - 1;
            x = 6 - x - y;
            goto Top;
        }
    }
}

int main(void)
{
    int n; // 円盤の枚数

    printf("円盤の枚数: ");
    scanf("%d", &n);

    move(n, 1, 3);

    return 0;
}
```