

Scratchによる アルゴリズム入門

第4回 再帰

2017.04.01

鎌倉シチズンネット(KCN)

再帰 (1)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

☆ユークリッドの互除法 (最大公約数を求めるアルゴリズム)

再帰アルゴリズムの例として無料体験会ではハノイの塔と階乗を学びましたが、ここではまず最大公約数を求める問題を扱います。最大公約数を求めるアルゴリズムとしては「ユークリッドの互除法」が有名です。

ユークリッドの互除法: 「 a と b の最大公約数は、 $a \div b = Q$ 余り r とすると、 b と r の最大公約数でもある。ただし、 $a > b > 0$ とする。」これを余り r が 0 になるまで繰り返し、そのときの b が最大公約数となる。

[問題]

☆二つの数字 a , b をキーボードから入力し、最大公約数を求める。

実行例:

自然数 a の値を入力してください 252

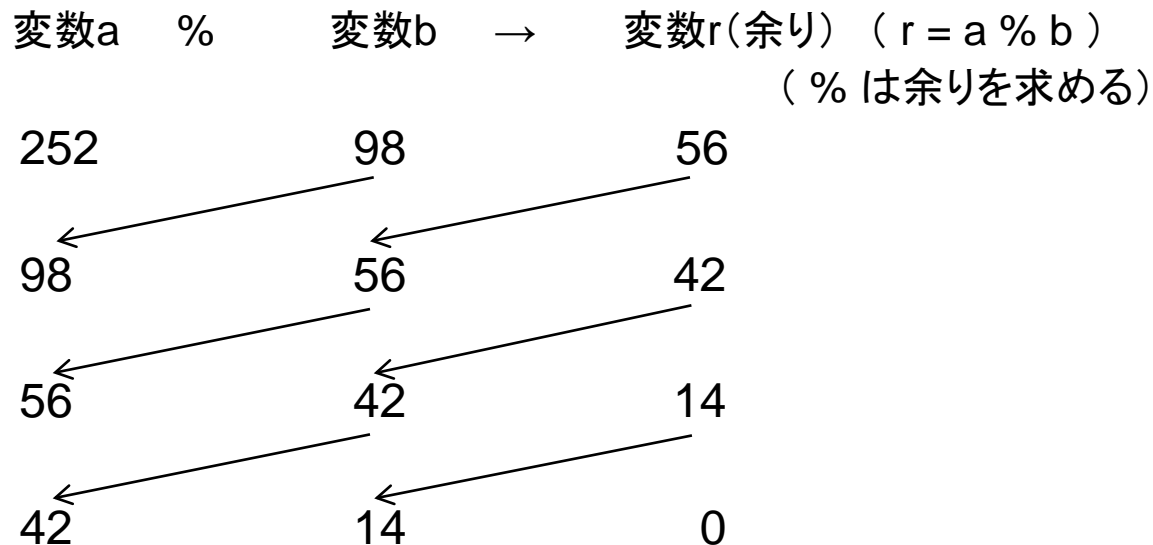
自然数 b の値を入力してください 98

252 と 98 の最大公約数は 14 です。

再帰 (2)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

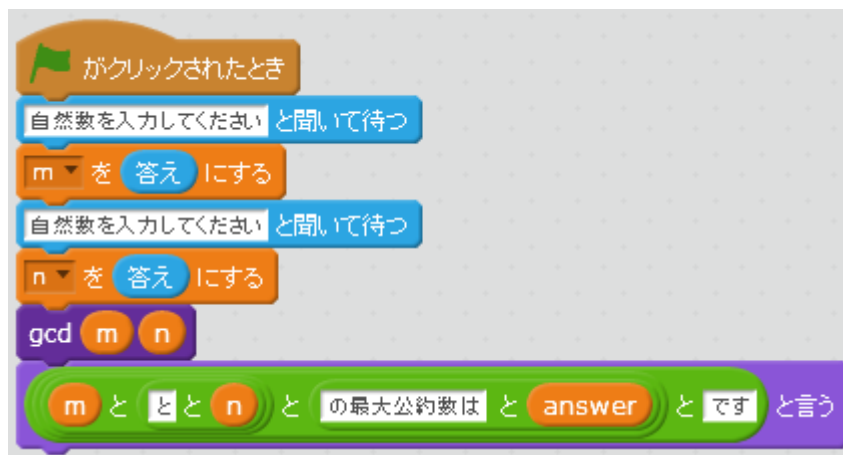
[補足] ユークリッドの互除法



余りが 0 でない間繰り返し、余りが 0 になったときの変数b の値が最大公約数になる。(最大公約数が 1 のとき、a と b は互いに素であるという)

再帰(3)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)



下記のプロジェクトは繰り返しを用いていますが、再帰を用いるようにリミックスしてください。

<https://scratch.mit.edu/projects/153584647/>

再帰(4)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

[参考]

C言語で記述すると次のようになります。

```
int gcd( int a, int b ) {  
    int r;  
    r = a % b;  
    if ( r == 0 ) {  
        return b;  
    }  
    return gcd( b, r );  
}
```

再帰(5)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

☆順列

n個の数字を使ってできる順列をすべて求めるプログラムを再帰関数を用いて作成します。

実行例:

数字の個数を入力してください。 3

123 132 213 231 321 312

☆注意

Scratchの関数には局所(ローカル)変数がないため、再帰関数を作成する場合には注意が必要である。配列(リスト)を利用して、「スタック」を用いなければならない場合がある。

(続く)

再帰(6)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

☆順列を求めるアルゴリズム

1, 2, ..., n個の順列の問題は、1~nを先頭にする n - 1個の順列の問題に分解することができる。先頭に1~nの値をもってくるには、数列の第1項と、第1項から第n項のそれぞれを逐次交換することにより行う。1, 2, 3, 4の順列は以下の4つの順列の問題に分解できる。

1 2 3 4 のうちの 2 3 4 の順列の問題に分解

2 1 3 4 のうちの 1 3 4 の順列の問題に分解

3 2 1 4 のうちの 2 1 4 の順列の問題に分解

4 2 3 1 のうちの 2 3 1 の順列の問題に分解

perm (i) は先頭から i 番目以降の順列を生成する再帰関数である。

① i 番目の値と j 番目の値を交換する

② perm (i + 1) の呼び出しを行う。

③ i 番目の値と j 番目の値を元に戻す

上記の処理を j = i から j = n になるまで繰り返す。

再帰(7)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

Scratch code for the main program:

- がクリックされたとき
- 数列の長さを入力してくださいと聞いて待つ
- n を 答え にする
- 左側に結果が表示されるのでよく見て!! と言う
- すべて 番目を permArray から削除する
- すべて 番目を jArray から削除する
- i を 1 にする
- n 回繰り返す
- i を 最後 番目に挿入する(permArray)
- i を 1 ずつ変える
- perm 1
- おしまい!! と言う

下記のプロジェクトの未完成部分をリミックスしてください。

<https://scratch.mit.edu/projects/153624407/>

Scratch code for the recursive function 'perm':

- 定義 perm i
- i を 最後 番目に挿入する(jArray)
- もし i < n なら
- 最後 番目(jArray) > n まで繰り返す
- 交換 i
- perm i + 1
- 交換 i
- 最後 番目(jArray) を 最後 番目(jArray) + 1 で置き換える
- でなければ
- リスト permArray を表示する
- 2 秒待つ
- リスト permArray を隠す
- 最後 番目を jArray から削除する

再帰(8)

- [アルゴリズムの基本 > 第3回 再帰 > Page 1 最大公約数](#)

[解答例]

最大公約数

<https://scratch.mit.edu/projects/153588125/>

順列

<https://scratch.mit.edu/projects/153593236/>