

アルゴリズム入門 第4回

千葉 滋



1

質問

- 家庭学習の為に自分のmacにも、大学PCの様な環境を整えたいが、どうすれば良いのか分からない。

- iruby をインストール
<https://github.com/SciRuby/iruby>



質問

- SyntaxError が直りません
 - if $0.1*r \leq d \leq 0.2*r$, $0.3*r \leq d \leq 0.4*r$,
 $0.5*r \leq d \leq 0.6*r$, $0.7*r \leq d \leq 0.8*r$,
 $0.9*r \leq d \leq 1.0*r$
 - カンマ, ではなく、論理和 `||` と書くべき?
 - if $0.1*r \leq d \ \&\& \ 0.2 * r \ || \ 0.3 * r \leq d \ \&\& \ \dots$



エラーの用語

- method
 - 関数のこととってください
- NilClass
 - 無 (nil) な値を扱おうとしているときに出る



関数の値（関数の計算結果）

- 最後に計算した式の値が関数の値

```
def sphere(s)
  image = make2d(s,s)
  for y in 0..(s-1)
    :
  end
  sphere(s) = image # 間違い
end

def sphere(s)
  image = make2d(s,s)
  for y in 0..(s-1)
    :
  end
  image # 正しい
end
```



return 文（教科書範囲外）

- 関数の値を決定して、そこで計算を終える

```
def sphere(s)
  image = make2d(s,s)
  for y in 0..(s-1)
    :
  end
  return image # 正しい
end
```



引数の scope

- なぜ `distance(x,y,r,s)` なのかわかりません。
`distance(x,y,r,r)` のはずじゃないんですか？

```

• def b(s,x,y)
  r = s / 2
  d = distance(x, y, r, r)    # d=distance2(x,y,r) も可
  :
end

def distance(x, y, r, s)
  ((x - r) ** 2 + (y - s) ** 2) ** (0.5)
end

def distance2(x, y, r)
  ((x - r) ** 2 + (y - r) ** 2) ** (0.5)
end

```



反復計算

- 残念ながら Ruby（やたいていの言語）では

```

def sum(n)
  1 + 2 + 3 + 4 + ... + n
end

```

と書いてもエラーとなり動かない。

- + ... + などと書いても察してくれない



ではどうする？

- 例えば漸化式で定義すれば OK

```
def sum(n)
  if n > 1 then
    sum(n - 1) + n
  else
    1
  end
end
```

再帰的関数

$$sum(n) = \begin{cases} sum(n-1) + n & (n > 1) \\ 1 & (otherwise) \end{cases}$$



代入を用いた方法

- for を使う

```
def sum(n)
  s = 0
  for i in 1..n
    s = s + i    # s == s+i ではない
  end
  s
end
```

- 変数 s の値が計算が進むにつれて変化する



(続) 代入を用いた方法

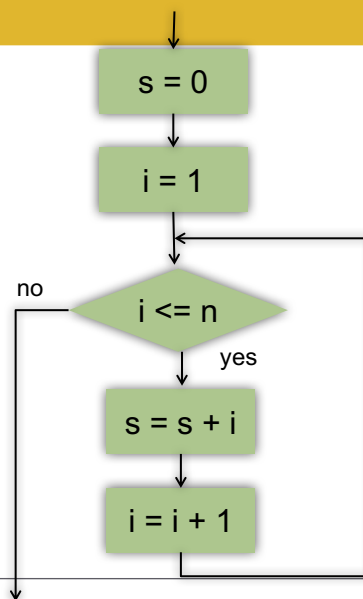
- while を使う

```
def sum(n)
  s = 0
  i = 1
  while i <= n do # i ≤ n の間 do..end 間を繰り返す
    s = s + i
    i = i + 1
  end
  s
end
```

do は省略可能



流れ図 (flow chart)



while の方が便利な場合

- Euclid の互除法による最大公約数の計算
 - for では難しい
- ```
def gcd(a, b)
 while b > 0 do
 oldA = a # a の古い値を保持
 a = b
 b = oldA % b # 剰余
 end
 a
end
```



## 再帰的関数の方がわかりやすいが...

- 負でない数  $a, b$  について、  
 $a$  を  $b$  で割った余りを  $r$  とすると、  
 $a$  と  $b$  の最大公約数は  $b$  と  $r$  のそれに等しい

```
def gcd(a, b)
 if b > 0 then
 gcd(b, a % b)
 else
 a
 end
end
```



## make2d 関数

- 縦 height 行、横 width 列の 2 次元配列を返す
  - 行列のようなものと考えればよい
- ```
def make2d(height, width)
  Array.new(height){ Array.new(width, 0) }
end
```
- `[[0, 0, ..., 0], [0, 0, ..., 0], ... [0, 0, ..., 0]]`
と書くより楽。



別の書き方

- 教科書 4.3.1 章
 - ```
def make1d(n)
 a = Array.new(n)
 for i in 0..(n - 1)
 a[i] = 0
 end
 a
end
```
  - ```
def make2d(h, w)
  a = Array.new(h)
  for i in 0..(h - 1)
    a[i] = make1d(w)
  end
  a
end
```



make1d を再帰関数で

- $[0] + [0]$ を計算すると $[0, 0]$
 $[1, 2] + [3, 4, 5]$ を計算すると $[1, 2, 3, 4, 5]$

ベクトル和ではない

- ```
def make1d(n)
 if n > 1 then
 make1d(n - 1) + [0]
 else
 [0]
 end
end
```



## 問題 1

- make2d も再帰的関数で書き直してみよ
- ヒント
  - make2d(1,w) の値 (計算結果) は、  
配列 [ make1d(w) ] となる。
  - make2d(1, 3) なら  
[ [0, 0, 0] ]  
つまり、[0, 0, 0] という要素を 1 つだけ含む配列
  - make2d(2, 3) なら  
[ [0, 0, 0],  
[0, 0, 0] ]



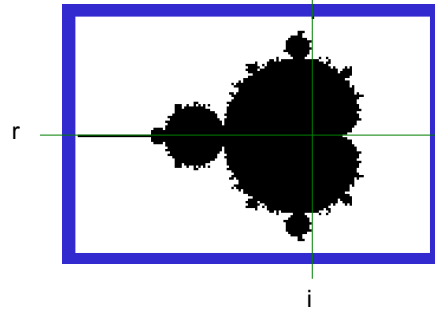
## Mandelbrot set

- 複素平面上の集合が作るフラクタル図形
  - 以下の複素数列  $\{z_n\}$  が発散しない複素数  $c$  の集合

$$\begin{cases} z_{n+1} = z_n^2 + c \\ z_0 = 0 \end{cases}$$

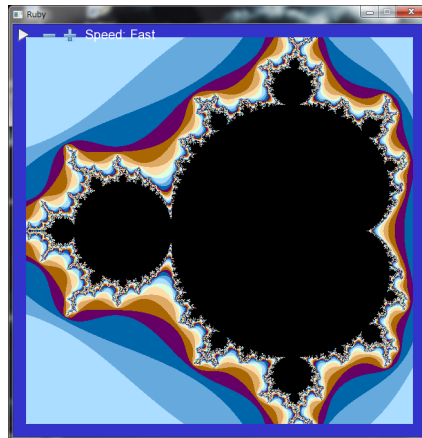
- 発散しない点が黒
- $(x, y)$  座標で表現すると

$$\begin{cases} r_{n+1} = r_n^2 - i_n^2 + x \\ i_{n+1} = 2r_n i_n + y \end{cases}$$



## 過去の作品

- マンデルブロ集合は有名なフラクタル図形



## 問題 2

- 前回の課題の関数  $b$  を変えて Mandelbrot set を描け。一部を拡大した図を描くとよい。
  - 発散の判定
    - $Z_{50}$  まで計算してみて、途中で  $|Z_n| > 2$  なら発散と判定
    - つまり  $x_n^2 + y_n^2 > 4$
  - $\text{image}[y][x] = b((x - 100) / 50.0, (y - 50) / 50.0)$  などとしないとちょうどよい範囲にならない
  - 色づけ
    - $|Z_n| > 2$  になる  $n$  の値で色を変えるとよい



## while を使う

- 反復を繰り返す条件に注意
  - ```
def b(x, y)
  n = 0
  r = 0.0
  i = 0.0
  while r**2 + i**2 < 4 && n <= 50 do
    oldR = r
    oldI = i
    n = ...
    r = ...
    i = ...
  end
  if n の値の条件 then 0 else 1 end
end
```



return を使う（教科書範囲外）

- 関数の計算を途中で強制的に打ち切って値を返す

```

• def b(x, y)
  n = 0
  r = 0.0
  i = 0.0
  while r**2 + i**2 < 4 do
    oldR = r
    oldI = i
    n = ...
    r = ...
    i = ...
    if n > 50 then
      return 0 # 計算を打ち切り b(x,y) の値を 0 とする
    end
  end
  while が終了した場合におこなう次の計算
end

```

&& n <= 50 がない



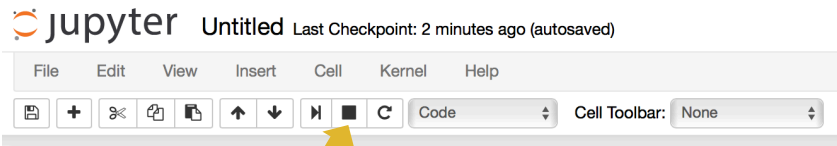
問題 3

- 退屈な人向け（普通の人はやらなくてもよい）
- Mandelbrot set を計算する関数 b を再帰的関数で書き直せ



念のため

- プログラムが途中で止まらなくなったら
 - (Shift + Return でプログラムが実行されなくなったら)



- 普通の ruby あるいは irb の場合
 - Control と C を同時に押して、それでもだめなら他のキーを押してみる



Error?

- iruby を起動すると
 - /usr/local/bin/iruby differs from registered path /opt/local/bin/iruby.
This might not work. Run 'iruby register --force' to fix it.
- ターミナルで下の命令を 1 回実行
 - iruby register --force

