

情報科学 04

- ・ 第3章 条件分岐と繰り返し
 - 条件分岐・**真偽値**・文字列・画像の作成
- ・ 第4章 関数から計算へ
 - **再帰による反復計算**
 - 繰り返し命令による反復計算

補足: includeを書く場所

```
include(Math)
```

数学関数を使う場合は
ファイルの先頭に書く

```
def heron(a,b,c)
```

```
include(Math)
```

関数の中に書いては
いけない

```
s = 0.5*(a+b+c)
```

```
sqrt(s * (s-a) * (s-b) * (s-c))
```

```
end
```

heron.rb

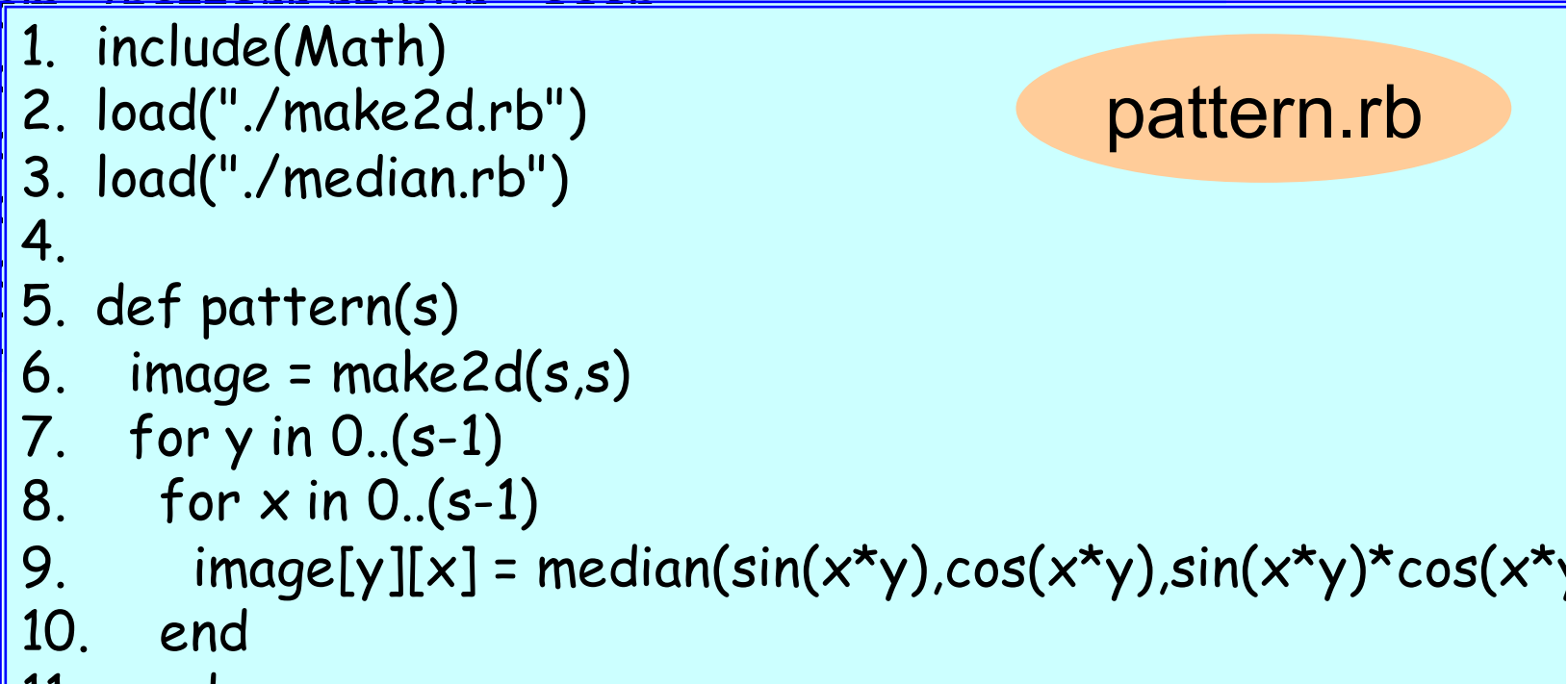
補足: 誤り報告の読み方 (p.14)

```
irb(main):009:0> load("./pattern.rb")  
true  
irb(main):010:0> show(pattern(10))  
NoMethodError: undefined method `<' for true:TrueClass  
  from ./median.rb:2:in `median'  
  from ./pattern.rb:9:in `pattern'  
  from ./pattern.rb:8:in `each'  
  from ./pattern.rb:8:in `pattern'  
  from ./pattern.rb:7:in `each'  
  from ./pattern.rb:7:in `pattern'  
  from (irb):10  
irb(main):011:0>
```

補足: 誤り報告の読み方

```
irb(main):009:0> load("./pattern.rb")
true
irb(main):010:0> show(pattern(10))
NoMethodError: undefined method `<' for true:TrueClass
from ./median.rb:2:in `median'
from ./pattern.rb:9:in `pattern'
```

```
from ./pattern.rb:9:in `pattern'
```



```
1. include(Math)
2. load("./make2d.rb")
3. load("./median.rb")
4.
5. def pattern(s)
6.   image = make2d(s,s)
7.   for y in 0..(s-1)
8.     for x in 0..(s-1)
9.       image[y][x] = median(sin(x*y),cos(x*y),sin(x*y)*cos(x*y))
10.    end
11.  end
```

pattern.rb

補足: 誤り報告の読み方

```
irb(main):009:0> load("./pattern.rb")
true
irb(main):010:0> show(pattern(10))
NoMethodError: undefined method `<' for true:TrueClass
  from ./median.rb:2:in `median'
  from ./pattern.rb:9:in `pattern'
  from ./pattern.rb:9:in `pattern'
```

```
1. def median(x,y,z)
2.   if x < y < z
3.     y
4.   else
5.     if y < z < x
6.       z
7.     else
8.       x
9.     end
```

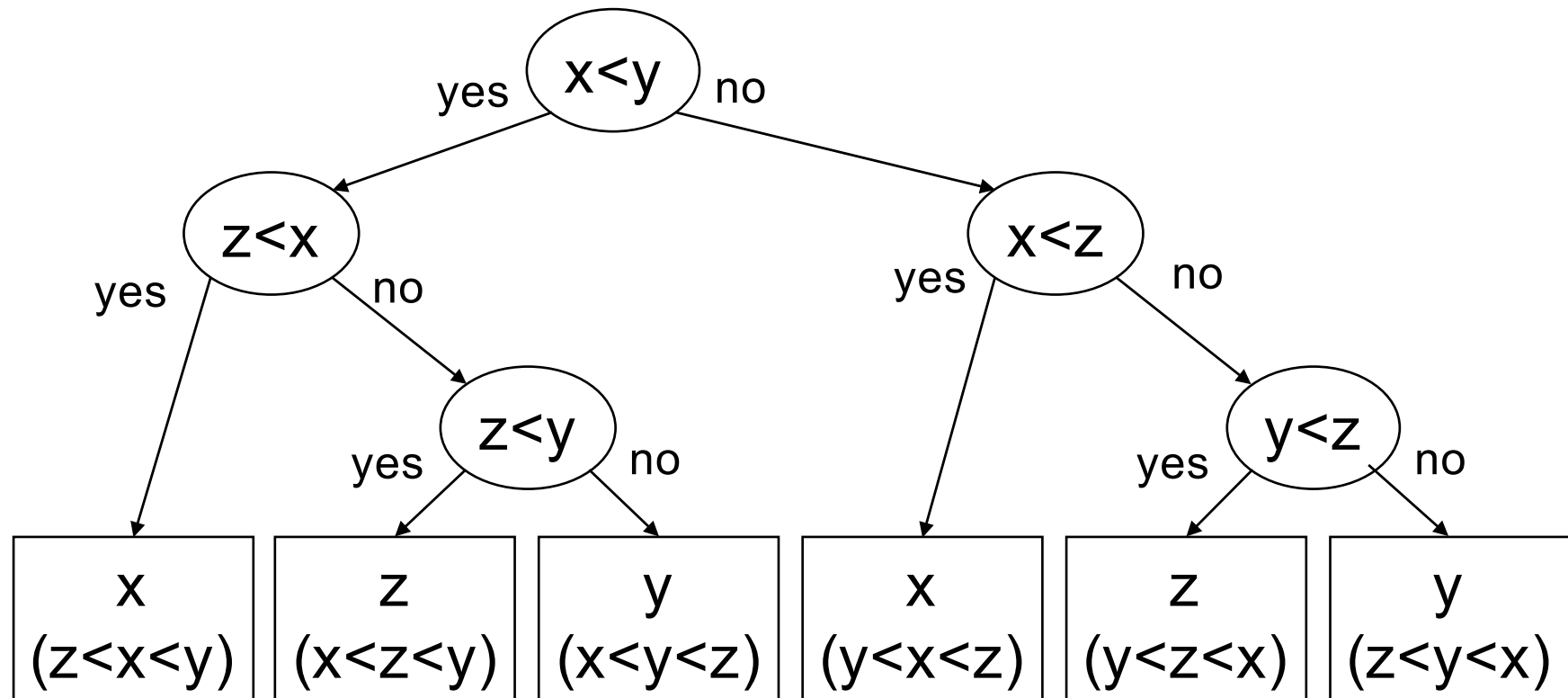
median.rb

```
1. include(Math)
2. load("./make2d.rb")
3. load("./median.rb")
4.
5. def pattern(s)
6.   image = make2d(s,s)
7.   for y in 0..(s-1)
8.     for x in 0..(s-1)
9.       image[y][x] = median(sin(x*y),cos(x*y),sin(x*
10.     end
```

pattern.rb

練習問題3.3c解説

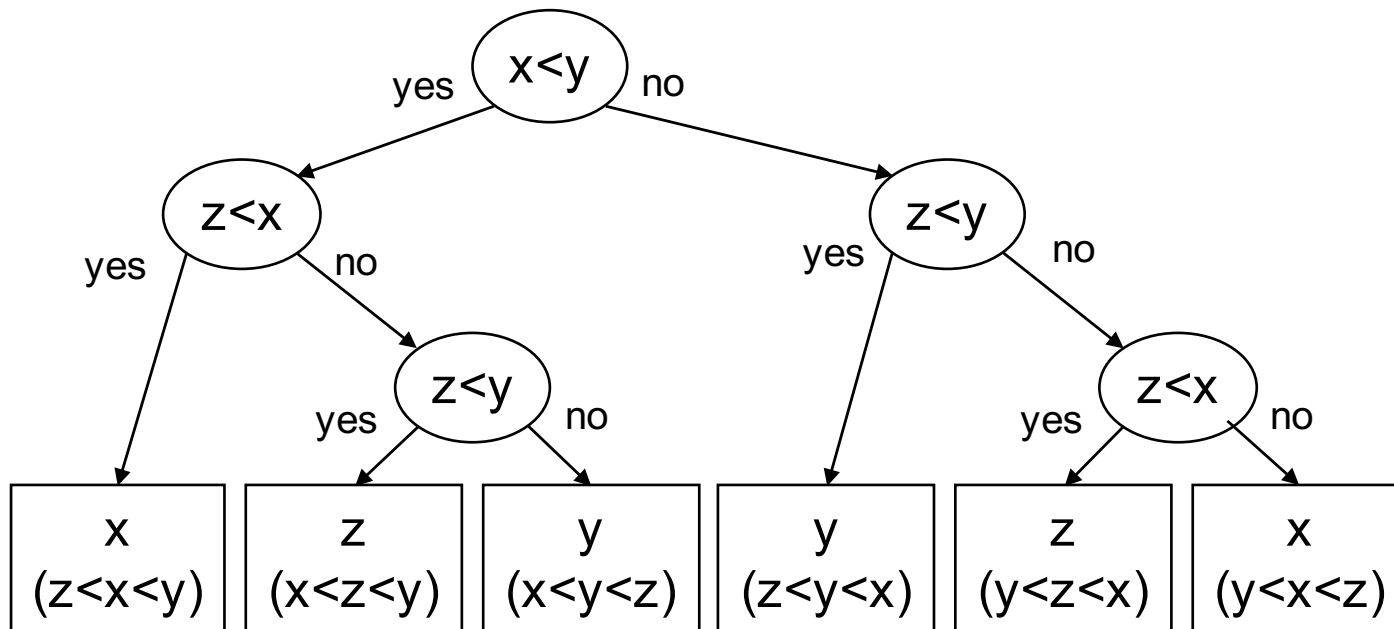
c) 3つの異なる値 x, y, z が与えられたときの中央値を求める $\text{median}(x, y, z)$.



練習問題3.3c解説

c) 3つの異なる値 x, y, z が与えられたとき、中央値を求める $\text{median}(x, y, z)$.

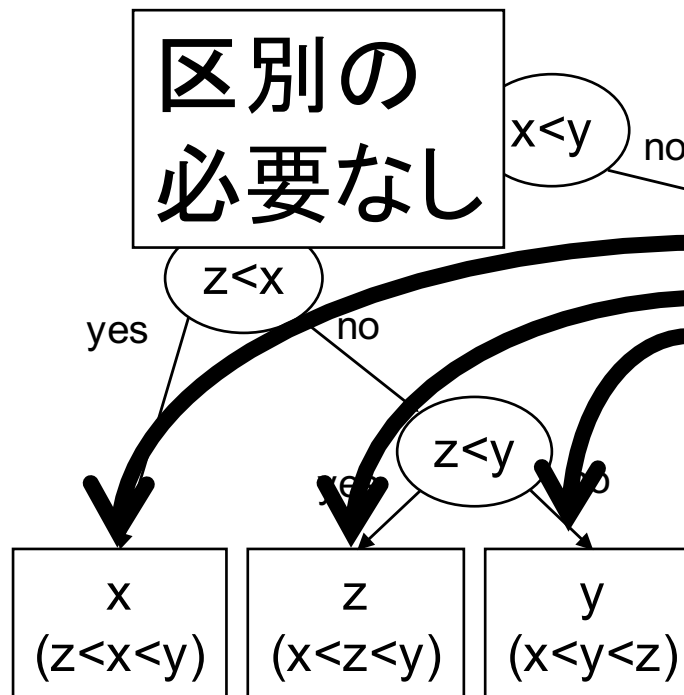
x, y, z の並び順は
 $3 \times 2 \times 1 = 6$ 通り



```
def median(x,y,z)
  if x < y
    if z < x
      x
    else
      if z < y
        z
      else
        y
      end
    end
  else
    if z < y
      y
    else
      if z < x
        z
      else
        x
      end
    end
  end
end
```

練習問題3.3c解説

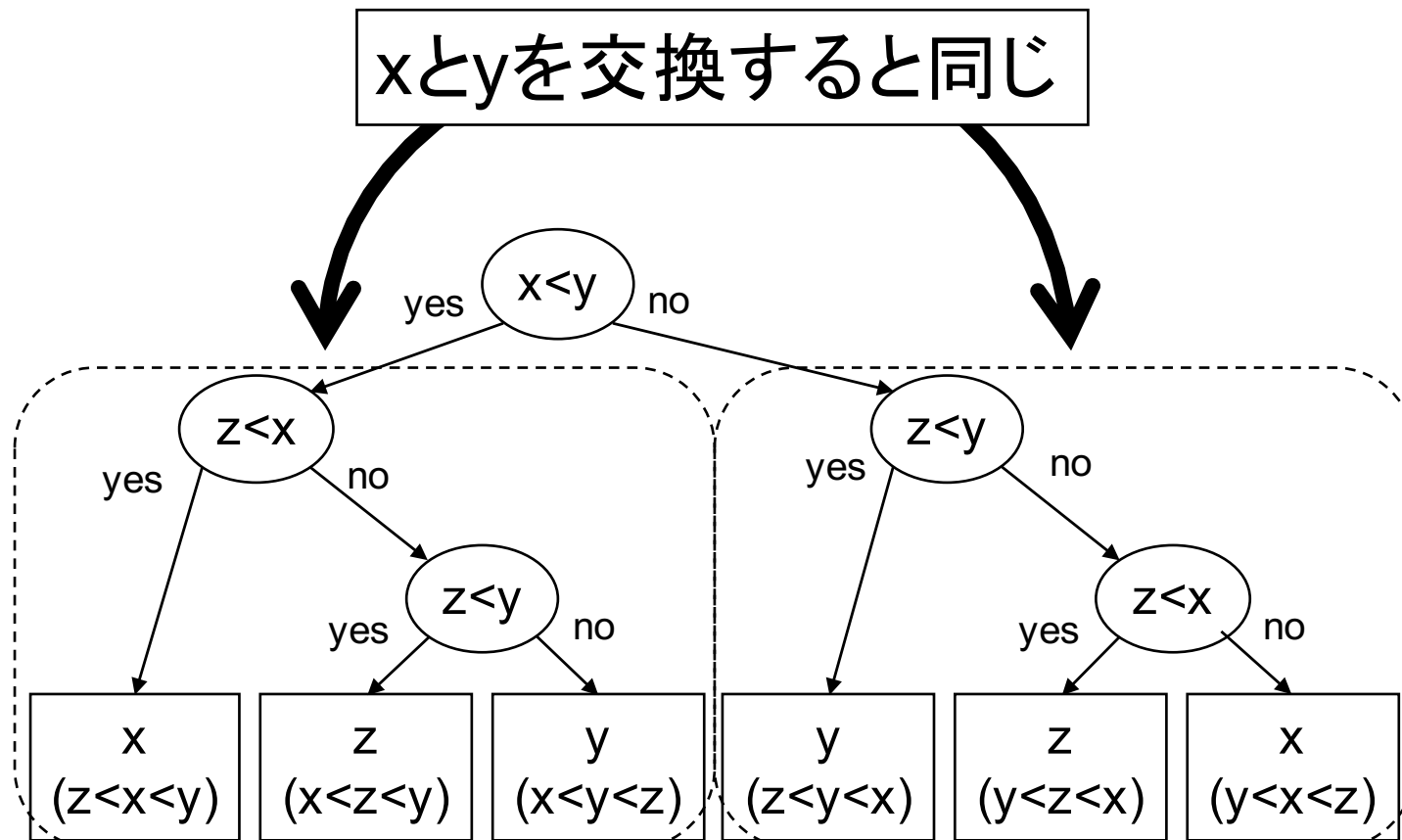
c) 3つの異なる値 x, y, z が与えられたときの中央値を求める $\text{median}(x,y,z)$.



```
def median (x,y,z)
  if x < y && y < z || z < y && y < x
    y
  else
    if y < z && z < x || x < z && z < y
      z
    else #z < x && x < y || y < x && x < z
      x
    end
  end
end
```

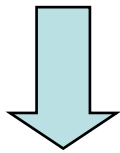
練習問題3.3c解説

c) 3つの異なる値 x, y, z が与えられたときの中央値を求める $\text{median}(x, y, z)$.



練習問題3.3c解説

c) 3つの異なる値 x, y, z が与えられたときの中央値を求める $\text{median}(x,y,z)$.



- $s < l$ のときに s, l, z の中央値を求める $\text{median1}(s, l, z)$
- s は x, y の小さい方
- l は x, y の大きい方

```
def median1(s,l,z)
  if z < s
    s
  else
    if z < l
      z
    else
      l
    end
  end
end
```

```
def median(x,y,z)
  if x < y
    median1(x,y,z)
  else
    median1(y,x,z)
  end
end
```

```
def median(x,y,z)
  if x < y
    if z < x
      x
    else
      if z < y
        z
      else
        y
      end
    end
  else
    if z < y
      y
    else
      if z < x
        z
      else
        x
      end
    end
  end
end
```

補足: 計算の進み方

- ・ $h=188.0$, $w=104.0$ のとき
 $w/(h/100.0)**2$ はどう計算されるか?

$w/(h/100.0)**2$

i. $w \rightarrow 104.0$ に置きかわる

ii. $(h/100.0)**2$

A) $h/100.0$

① $h \rightarrow 188.0$

B) $188.0/100.0 \rightarrow 1.88$

$1.88**2 \rightarrow 3.5344$

$104.0/3.5344 \rightarrow 29.45250\dots$

- ・ 全体の式の計算は
 - ・ 直下の部分式を計算
 - ・ 求められた値で置き換える
- ・ 変数はしまわれている値で置き換える
- ・ 数値だけの式を計算

補足: 計算の進み方

bmi(180.0+8,sqrt(10816))はどう計算されるか?

1. bmi(180.0+8,sqrt(10816))

180.0+8 → 188.0

sqrt(10816) → 104.0

2. h=188.0, w=104.0
として

3. $w / (h / 100.0) ** 2$
を計算(以下略)

```
def bmi(h, w)
  w / (h / 100.0) ** 2
end
```

• 関数呼出の計算は

• 各引数の式を計算

• 関数定義中の変数に
求められた値をしまい

• 関数定義の本体の式を計算

3.2 真偽値を与える論理演算

irb(main):003:0> $x = 3$

$\Rightarrow 3$

irb(main):004:0> $1 < x$

$\Rightarrow \text{true}$

irb(main):005:0> $x == 2$

$\Rightarrow \text{false}$

```
def is_even(x)
```

```
  x%2 == 0
```

```
end
```

「これは偶数か？」
という関数

is_even.rb

```
load("./is_even.rb")
```

```
def tnp(n)
```

```
  if is_even(n)
```

```
    n/2
```

```
  else
```

```
    3*n + 1
```

```
  end
```

```
end
```

条件だけを取り出して

関数にすることができる

- これは素数か?
- この年は閏年か?
- これは「ほとんどゼロ」か?
- x,y,zは小さい順に並んでいるか?
- xはyで割り切れるか?
- この点はこの直線上にあるか?
- :

tnpo.rb⁴

論理演算

y年2月の日数を求める関数

```
def days_of_february(y)
  if y年は閏年?
    29
  else
    28
  end
end
```

y年の日数を求める関数

```
def days_of_year(y)
  if y年は閏年?
    366
```

Y年m月の日数を求める関数

```
def days_of_month(y,m)
  if m==2 && y年は閏年?
    29
  else
    if m==1 || m==3 ||...
```

なぜ $x < y < z$ と書けないのか？

$x=1, y=2, z=3$ のとき

- $x < y < z$ は $(x < y) < z$ と同じ
 1. $x < y$ を計算
 2. z を計算
 3. < を計算

```
1. def median(x,y,z)
2.   if x < y < z
```

```
irb(main):010:0> show(pattern(10))
```

```
NoMethodError: undefined method `<' for true:TrueClass
```

```
from ./median.rb:2:in `median'
```

```
0.     z
7.   else
8.     x
9.   end
```

$z < x$

第4章 関数から計算へ (pp.51-)

4.1 繰り返しによる反復計算

4.2 **再帰による反復計算**

4.3 配列・文字列と繰り返し

「関数」だけで「計算」ができる

- 関数:

- 「 $ax^2 + bx + c = 0$ の実解」(数学関数)
- 「 x と y の大きい方」(条件分岐)

- 計算:

- 「1から n までの和」
- 「 n と m の最大公約数」
- 「 n は素数か?」
- 「『 x から $(x^2-2)/2x$ を引く』ことを繰り返したときの x の収束値」

```
def max(x)
  if x<y
    y
  else
    x
end
```

4.2 再帰による反復計算

- 1からnまでの和

$$\text{sum}(n) = 1+2+\dots+n$$

- $\text{sum}(n)$ を $\text{sum}(n-1)$ で表わすと

$$\text{sum}(n) = \text{sum}(n-1) + n \quad (n \geq 2)$$

$$\text{sum}(n) = 1 \quad (n=1)$$

```
def sum(n)
  if n >= 2
    sum(n-1) + n
  else
    1
  end
end
sum.rb
```

再帰による計算をもっと

定義ができた人は
完全数を探してみよう!

完全数とは、自身を除く
約数の和が自身に一致
する数

- 1からnまでの数のうち
kの約数の和 $sod(k,n)$
- $sod(k,n)$ を $sod(k,n-1)$ で
 - 場合分け: nがkの約数か・
約数でないか・nが1のとき

$$sod(k,n) = \left\{ \begin{array}{l} \dots \\ \dots \\ \dots \end{array} \right.$$

```
def sod(k,n)
  if n >= 2
    if divisible(k,n)
      sod(k,n-1)+n
    else
      sod(k,n-1)
    end
  else
    1
  end
end
```