情報Ⅰ

4-1-14　数学・科学用・統計用ライブラリ紹介

（英語における文法・構文のような定型コーディング）

・統計用ライブラリstatistics

・科学技術計算の基礎パッケージライブラリnumpy

・データ可視化ライブラリmatplotlib

・数学用モジュールmath，cmath

・探索、並べ替え、方程式解法比較

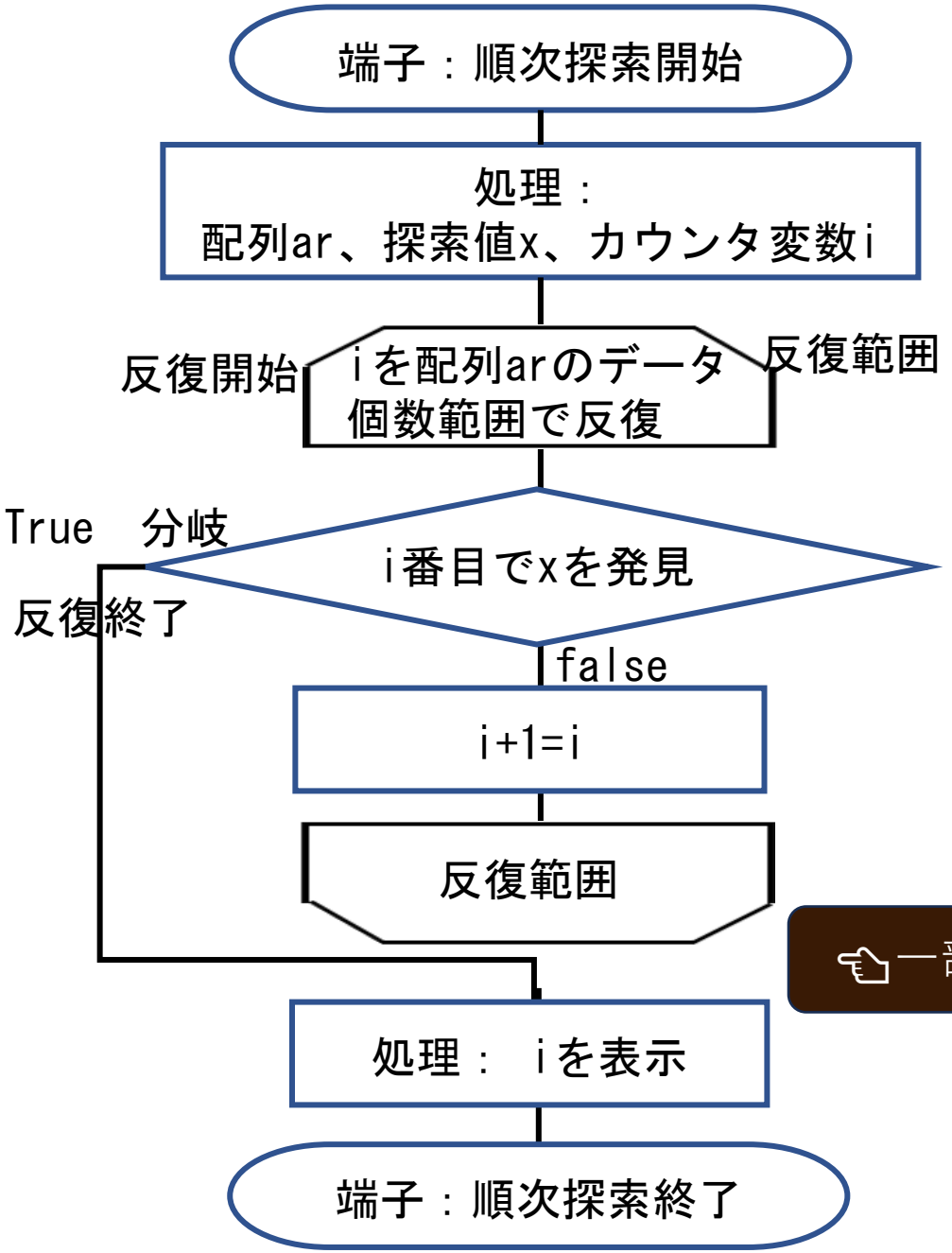
定型文
There is a key in my pocket.

試料PDFをテキストコピーしてコーディングしてください

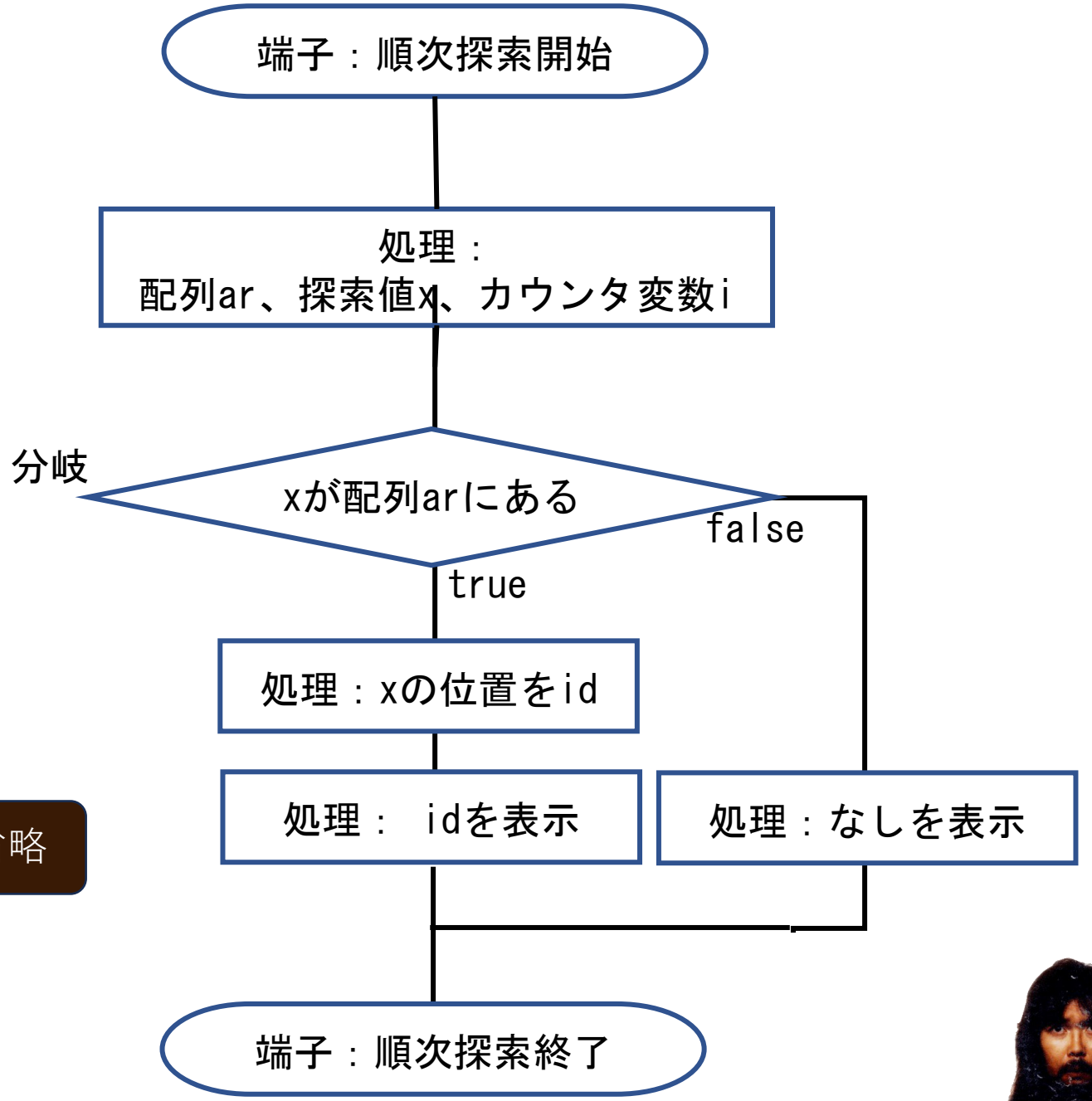
教科書・共通テスト
➡算法（思考）
ステップ数（行数）大

統計分析・アプリ開発
➡文法（定型文）
ステップ数（行数）小

There are some books on the table.


端子：順次探索開始
処理：
配列ar、探索値x、カウンタ変数i
反復開始
iを配列arのデータ個数範囲で反復
反復範囲
True　分岐
i番目でxを発見
反復終了
false
i+1=i
反復範囲
処理：　iを表示
端子：順次探索終了

☜一部省略


端子：順次探索開始
処理：
配列ar、探索値x、カウンタ変数i
分岐
xが配列arにある
false
true
処理：xの位置をid
処理：　idを表示
処理：なしを表示
端子：順次探索終了

(11-6) ①アルゴリズムに基づく順次探索サンプル1

```
ar = [3,8,6,1,9,5,4,2,7]      # 配列要素
x=1                           # ターゲットを x=1とする
for i in range(len(ar)):      # i は配列データの個数回だけ反復
  if ar[i] == x:              # 配列 i 番目がターゲットならば
    print("INDEX", i)         # "INDEX" i を表示して
    break                     # 反復終了
```

カウント変数 i	0	1	2	3	4	5	6	7	8
配列要素	3	8	6	1	9	5	4	2	9

```
ar = [3, 8, 6, 1, 9, 5, 4, 2, 7]
x=1
for i in range(len(ar)):
  if ar[i] == x:
    print("INDEX", i)
    break
```

```
INDEX 3
```

（11-6） ①アルゴリズムに基づく順次探索サンプル2

```
ar = [3,8,6,1,9,5,4,2,7]   # 配列要素
x=2                        # ターゲット x=2
if x in ar:                # xが配列にあれば
  id=ar.index(x)           # 変数idをxを発見したカウンタ数とする
  print("INDEX", id)       # "INDEX" id を表示して
else:                      # xが配列に無ければ
  print('none')            # "none"を表示
```

カウント変数 i	0	1	2	3	4	5	6	7	8
配列要素	3	8	6	1	9	5	4	2	9

```
ar = [3,8,6,1,9,5,4,2,7]
x=2
if x in ar:
  id=ar.index(x)
  print("INDEX", id)
else:
  print('none')
```

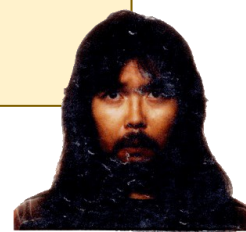
```
INDEX 7
```

```
ar = [3,8,6,1,9,5,4,2,7]
x=21
if x in ar:
  id=ar.index(x)
  print("INDEX", id)
else:
  print('none')
```

```
none
```

（11-6） ④アルゴリズムに基づく二分探索　　サンプル1

```
arr = [3,8,6,1,9,5,4,2,7]          # 配列array
x=7                                 # ターゲットを x=7とする
arr.sort()                          # モジュールで昇順並べ替え  arr = [1,2,3,4,5,6,7,8,9]になる
left = 0                            # 配列左端は left = 0 とする          0番のデータは1
right = len(ar) - 1                 # 配列右端は right =データ個数-1   8番のデータは9
while left <= right :               # left <= rightの間反復
  m = (left + right) // 2           # 配列中央は左端+右端の2分の1切り捨て  すなわち4番目
  if arr[m] == x :                  # 配列中央がターゲット x=7なら
    print("INDEX",m)                # "INDEX",m を表記して
    break                           # 反復終了
  elif arr[m] < x:                  # 配列中央データがターゲット より小さければ・・・本ケース
    left = m + 1                    # 探索する左端を中央+1にする・・・3行目#      探索
  else :                            # 配列中央データがターゲットより大きければ
    right = m -1                    # 探索する右端を中央-1にする
```



```
arr = [3,8,6,1,9,5,4,2,7]
x=7
arr.sort()
left = 0
right = len(arr)-1
while left <= right :
  m = (left + right) // 2
  if arr[m] == x:
    print("INDEX",m)
    break
  elif arr[m] < x:
    left = m + 1
  else :
    right = m -1
```

```
INDEX 6
```

カウント変数 i	0	1	2	3	4	5	6	7	8
配列要素	3	8	6	1	9	5	4	2	9

カウント変数 i	0	1	2	3	4	5	6	7	8
配列要素	1	2	3	4	5	6	7	8	9

```
def src(arr, x):
  arr = [3,8,6,1,9,5,4,2,7]
  x=7
  arr.sort()
  l = 0
  r = len(arr) - 1
  while l <= r:
    m = (l + r)//2
    if arr[m] == x:
      return m
      break
    elif arr[m] < x:
      l = m + 1
    else:
      r = m - 1
  return "none"
result = src(arr, x)
print('INDEX',result)
```

サンプル2
サンプル1関数化
アルゴリズムは同義

アルゴリズム実装二分探索
17～18行・・・共通テスト受験向け

```python
def src(arr, x):
  arr = [3,8,6,1,9,5,4,2,7]
  x=7
  arr.sort()
  l = 0
  r = len(arr)- 1
  while l <= r:
    m = (l + r) // 2
    if arr[m] == x:
      return m
      break
    elif arr[m] < x:
      l = m + 1
    else:
      r = m - 1
  return "none"
result = src(arr, x)
print('INDEX',result)
```

```
INDEX 6
```

```python
def src(arr, x):
  arr = [3,8,6,1,9,5,4,2,7]
  x=71
  arr.sort()
  l = 0
  r = len(arr)- 1
  while l <= r:
    m = (l + r) // 2
    if arr[m] == x:
      return m
      break
    elif arr[m] < x:
      l = m + 1
    else:
      r = m - 1
  return "none"
result = src(arr, x)
print('INDEX',result)
```

```
INDEX none
```

(11-6) ⑥モジュールを活用した探索

SQL文と同等で簡便！

```
list = [3,8,6,1,9,5,4,2,7]
result =4 in list
print(result)
index = list.index(4)
print(index)
```

値の有無

カウンタ

モジュール活用探索

3行！

```
list = [4,5,2,1,3,5,3,5,1]
cnt = list.count(5)
print(cnt)
x=max(list)
print(x)
y=min(list)
print(y)
a=sum(list)
print(a)
b=len(list)
print(b)
```

値の個数

最大

最小

値の合計

データ数

(11-7) ①アルゴリズムに基づく文字列探索

```
list= ['albert','carol','ester','jane','peter','george','william','Catherine','john','richard','Diana']
new_list =[ ]
for x in list:
    if "er" in x:
        new_list.append(x)
print(new_list)
```

アルゴリズム実装
6行

(11-7) ②リスト内包表記に基づく文字列探索

```
list= ['albert','carol','ester','jane','peter','george','william','Catherine','john','richard','Diana']
match = [x for x in list if "er" in x]
print(match)
```

リスト内包表記活用
3行

(11-5) ① アルゴリズムに基づくバブルソート昇順

```
def bs(ar):                                  # 関数定義bubble sort(array)
  n = len(ar)                                # 変数nを配列データ個数
  for i in range(n):                         # カウンタ変数iを0からn回反復
    for j in range(0, n-i-1):                # カウンタ変数jを0からn-i-1まで反復
      if ar[j] > ar[j+1]:                    # 隣接する2数の左側が大きい場合
        ar[j], ar[j+1] = ar[j+1], ar[j]      # 隣接する2数の順を入れ替え
ar = [3,8,6,1,9,5,4,2,7 ]                    # 配列array
bs(ar)                                       # 関数処理bubble sort(array)
print("result", ar)
```

アルゴリズム実装バブルソート
9行・・・共通テスト受験向け

```python
def bs(ar):
    n = len(ar)
    for i in range(n):
        for j in range(0, n-i-1):
            if ar[j] > ar[j+1]:
                ar[j], ar[j+1] = ar[j+1], ar[j]
ar = [3,8,6,1,9,5,4,2,7 ]
bs(ar)
print("result", ar)
```

```
result [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

(11-5) ① アルゴリズムに基づくバブルソート降順

```
def bs_desc(ar):
  n = len(ar)
  for i in range(n-1):
    for j in range(0, n-i-1):
      if ar[j] < ar[j+1]:
        ar[j], ar[j+1] = ar[j+1], ar[j]
ar = [3,8,6,1,9,5,4,2,7]
bs_desc(ar)
print("newarray", ar)
```

アルゴリズム実装バブルソート

9行

```python
def bs_desc(ar):
    n = len(ar)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if ar[j] < ar[j+1]:
                ar[j], ar[j+1] = ar[j+1], ar[j]
ar = [3,8,6,1,9,5,4,2,7]
bs_desc(ar)
print("newarray", ar)
```

```
newarray [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

(11-5) ② リスト内包表記活用ソート

```
list = [3,8,6,1,9,5,4,2,7]
list.sort()
print(list)
```

```
list = [3,8,6,1,9,5,4,2,7]
list.sort(reverse=True)
print(list)
```

リスト内包表記活用ソート

3行！

```
list = [3, 8, 6, 1, 9, 5, 4, 2, 7]
list.sort()
print(list)
```

```
list = [3, 8, 6, 1, 9, 5, 4, 2, 7]
list.sort(reverse=True)
print(list)
```

(19) 数学用モジュールmath　…表計算ソフトウェアの関数的な処理

```
import math                          # mathモジュールの導入
print(math.pi)                       # 円周率π
x=6.25                               # x＝6.25
print(math.ceil(x))                  # 切り上げ
print(math.sqrt(x))                  # 平方根
print(math.gcd(225,150))             # 最大公約数
y=10000                              # y=10000
print(math.log10(y))                 # 対数　　本例log1010000=4
kakudo=180                           # kakudo=180°（度数法）
print(math.radians(kakudo))          # kakudoを度数法➡弧度法変換　本例180°＝π
print(math.sin(math.pi/6))           # 三角関数sin
print(math.cos(math.pi/6))           # 三角関数cos
print(math.tan(math.pi/6))           # 三角関数tan
```

```
3.1415
7
2.5
75
4.0
3.1415
0.4999
0.8660
0.5773
```

(26) 代数計算・数式処理ライブラリsympy

(26-1) 関数の2変数に数値代入

例題　$f(x,y) = x^2 + \sqrt{x} + y^3 - 70$　　$f(9,2)=?$

```
import sympy                                  # sympyを定義
x = sympy.Symbol('x')                         # 変数xを定義
y = sympy.Symbol('y')                         # 変数yを定義
f = x**2 + sympy.sqrt(x)+ y**3 -79            # 関数　f=x²+y²−9
print(f.subs([(x, 9), (y, 2)]))               # x=9、y=2を代入
```

```
import sympy
x = sympy.Symbol('x')
y = sympy.Symbol('y')
f = x**2+sympy.sqrt(x) + y**3 -70
print(f.subs([(x, 9), (y, 2)]))
```

```
22
```

(26) 代数計算・数式処理ライブラリsympy

(26-2) 関数定義と展開

例題　y = (x + 2)(x − 3)　を展開

```
import sympy
x = sympy.Symbol('x')          # 変数xを定義
y = (x + 2)*(x - 3)            # 元式を定義
print(sympy.expand(y))         # 展開
```

```
import sympy
x = sympy.Symbol('x')
y = (x + 2)*(x - 3)
print(sympy.expand(y))
```

```
x**2 - x - 6
```

(26-3) 関数定義と因数分解

例題　$y=x^2-x-6$ を因数分解

```
import sympy
x = sympy.Symbol('x')          # 変数xを定義
y = x**2 - x - 6               # 元式を定義
f= sympy.factor(y)             # 因数分解
print(f)
```

または

```
from sympy import *
x = Symbol('x')
y = x**2 - x - 6
print(factor(y))
```

```
import sympy
x = sympy.Symbol('x')
y = x**2 - x - 6
f= sympy.factor(y)
print(f)
```

```
(x - 3)*(x + 2)
```

(26-4) 関数定義と微分

例題　$y=x^2-3x-6$ を微分

```
import sympy
x = sympy.Symbol('x')
y = x**2 - 3*x - 6
print(sympy.diff(y))
```

```
# 変数xを定義
# 元式を定義　y=x^2-3x-6
# 微分
```

または

```
from sympy import *
x = Symbol('x')
y = x**2 - 3*x - 6
print(diff(y,x))
```

```
import sympy
x = sympy.Symbol('x')
y = x**2 - 3*x - 6
print(sympy.diff(y))
```

```
2*x - 3
```

(26-5) 関数定義と積分

例題　y = x**2 - x - 6　を不定積分、x=3～4の範囲で定積分

```
import sympy
x = sympy.Symbol('x')                    # 変数xを定義
y = x**2 - x - 6                         # 元式を定義　y=x2-x-6
print(sympy.integrate(y))                # 不定積分
```

```
x**3/3 - x**2/2 - 6*x
```

```
import sympy
x = sympy.Symbol('x')
y = x**2 - x - 6
print(sympy.integrate(y,(x,3,4)))        # 定積分
```

```
17/6
```

(26-6) 二次方程式解法2次方程式

例題　$x^2-3x+2=0$ を解く

```
import sympy
x = sympy.Symbol('x')                    # 変数、式を定義
s=sympy.solve(x**2 - 3 * x + 2)     # 方程式を解く
print(s)
```

```
import sympy
x = sympy.Symbol("x")
s=sympy.solve(x**2 - 3 * x + 2)
print(s)
```

```
[1, 2]
```

(26-7) 連立方程式解法

例題　$3x+5y=29$

$x+y=7$

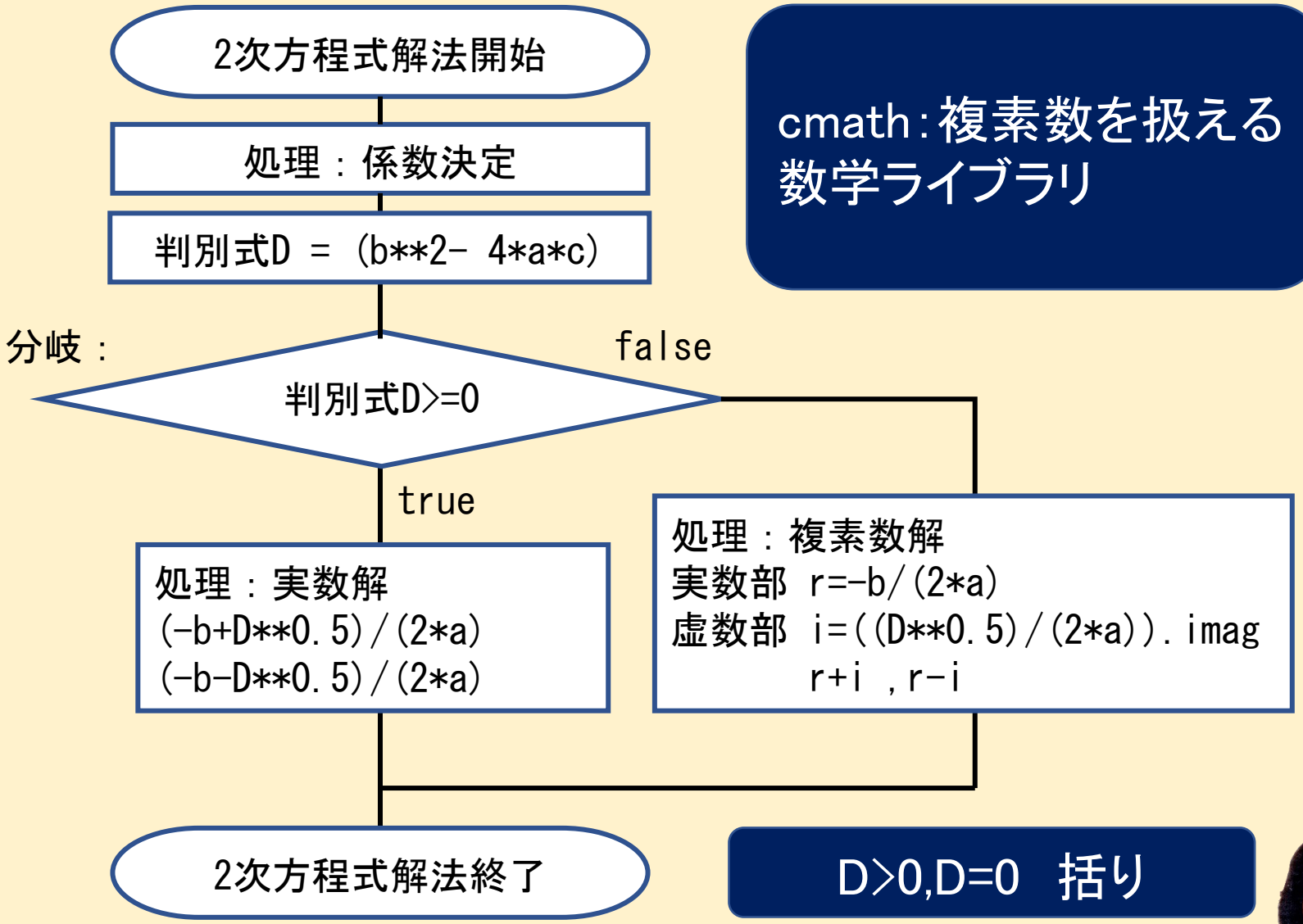
```
import sympy                  # 代数計算　数式処理ライブラリ
x = sympy.Symbol('x')    # 変数xを定義
y = sympy.Symbol('y')    # 変数yを定義
f = 3 * x + 5 * y - 29    # 関数 f =3x+5y-29
g = x + y - 7                  # 関数 g =x+y-7
print(sympy.solve([f, g]))  #  f=0、g=0の方程式を解く
```

```
import sympy
x = sympy.Symbol('x')
y = sympy.Symbol('y')
f = 3 * x + 5 * y - 29
g = x + y - 7
print(sympy.solve([f, g]))
```

```
{x: 3, y: 4}
```

(13-4) 2次方程式における解の公式をFCDから実装・・・数Ⅱ既習

```
import cmath
def g(a,b,c):
  D = (b**2- 4*a*c)
  If D>0 or D==0:
    p = (-b+D**0.5)/(2*a)
    q = (-b-D**0.5)/(2*a)
    return p,q
  else:
    r= -b/(2*a)
    i= ((D**0.5)/(2*a)).imag
    p= complex(r, i)
    q= complex(r, -i)
    return p,q
a=float(input("a"))
b=float(input("b"))
c=float(input("c"))
print(g(a,b,c))
```



cmath：複素数を扱える数学ライブラリ

D>0,D=0 括り

(26-8) 二つの関数の交点座標を求め、グラフで示す。

$y=x^2$

$y=2x+24$

```
import matplotlib.pyplot as plt  #グラフ描画ライブラリ
import numpy as np               # 統計用ライブラリ
import sympy                     # 代数計算　数式処理ライブラリ
x = sympy.Symbol('x')
y = sympy.Symbol('y')
f = x ** 2 - y                   # f= x^2 -y
g = 2 * x +24 - y                # g=2x+24-y
print(sympy.solve([f, g]))       # 方程式を解く

x = np.arange(-10, 10, 0.1)      # x座標(-10～+10)
plt.plot(x, x ** 2, c='blue')    # y= x^2　式のグラフ描画
plt.plot(x, 2 * x + 24 , c='red')  # y=2x+24　式のグラフ描画
plt.show()
```

```
[{x: -4, y: 16}, {x: 6, y: 36}]
```