アルゴリズム入門　共通問題（2015年度Aセメスター試験）

[試験日時：2015年12月25日第3限（60分），答案用紙：1部，計算用紙：1枚，持ち込み一切不可]

内容に関する質問は受け付けない．問題の記述が曖昧な場合は，適切な仮定をおいて回答し，どのような仮定をおいたのか明記せよ．

以下の問題で用いられている make2d(m，n) は $m \times n$ の2次元配列を作る．これは定義済みとしてよい．

問題1

トリボナッチ数列は以下の3項間漸化式で定義される．

$$T_0 = T_1 = 0, \quad T_2 = 1$$
$$T_i = T_{i-1} + T_{i-2} + T_{i-3} \ (i \geq 3).$$

トリボナッチ数列の第 n 項 T_n を計算したい．これに関して以下の問に答えよ．

(1) 図1の空欄 (A) ～ (C) を埋め，再帰を用いて T_n $(n \geq 0)$ を計算する関数 tri_rec を完成させよ．

(2) 図2の空欄 (D) ～ (F) を埋め，ループを用いて T_n $(n \geq 3)$ を計算する関数 tri_loop を完成させよ．

(3) 図3の空欄 (G) ・ (H) を埋め，3×3 行列の乗算を用いて T_n $(n \geq 3)$ を計算する関数 tri_mat を完成させよ．なお，matpower(a，n) は2次元配列 a を行列とみなしたときに，その n 乗を計算する関数である．

(4) 3つの関数 tri_rec, tri_loop, tri_mat を計算量の観点から比較せよ．

```
def tri_rec(n)
  if (A)
    0
  else
    if (B)
      1
    else
      (C)
    end
  end
end
```

図1: プログラム tri_rec

```
def tri_loop(n)
   p3 = 0
   p2 = 0
   p1 = 1
   c = 1
   for i in 4..n
     p3 = p2
     p2 = (D)
     p1 = (E)
     c = (F)
   end
   c
end
```

図2: プログラム tri_loop

```
def tri_mat(n)
   a = make2d(3, 3)
   for i in 0..2
     for j in 0..2
        if (G)
           a[i][j] = 1
        else
           a[i][j] = 0
        end
     end
   end
   b = matpower(a, (H))
   b[0][0]
end
```

図3: プログラム tri_mat

問題2

(1) 二次方程式 $ax^2 + bx + c = 0$（ただし $a \neq 0$）の解を公式 $x = (-b \pm \sqrt{b^2 - 4ac})/2a$ を用いて Ruby で計算することを考える．以下の問に答えよ．

1. $a = 1.0$，$b = -4.0$，$c = 3.0$ の場合と，$a = 0.1$，$b = -0.4$，$c = 0.3$ の場合は，本質的には同じ二次方程式であり，$x = 1$ が解の 1 つである．しかし，Ruby で計算してみたところ，前者に対しては `1.0`，後者に対しては `0.9999999999999996` という結果が得られた．この原因を述べよ．
2. $a = 0.5$，$b = -0.9999999$，$c = 0.4999999$ の場合を考える．このときは $x = 1$ が解の 1 つであるが，Ruby で計算してみたところ，`1.0000000005138276` と大きな誤差を含む結果が得られた．この原因を述べよ．
3. $a = c = 10^{-5}$，$b = 10^4$ の場合を考える．このとき，Ruby で計算してみたところ，解の 1 つが `0.0` と得られたが，$x = 0$ は解ではない．この原因を述べよ．

(2) 原点を中心とした半径 1 の四半円（四分円）の面積を 4 倍することで円周率を計算したい．このために，図 4 に示す，台形公式による数値積分プログラムを作成した．ここで `sqrt(`n`)` は n の平方根を求める関数である．以下の問に答えよ．

1. 空欄 (A) ・ (B) を埋めよ．
2. `pi(10)`，`pi(1000)`，`pi(100000)` の結果はそれぞれ `3.1045183262483182`，`3.141555466911023`，`3.141592616402008` であった．これらの値が円周率と異なる理由，および，引数 `n` を大きくすると円周率に値が近づいてゆく理由を説明せよ．
3. 引数 `n` が大きければ大きいほど，計算結果は円周率に近いだろうか．理由と共に述べよ．

```
include(Math)
def pi(n)
  delta = 1.0 / n
  s = 1.0 / 2.0
  for i in 1..(n - 1)
    s = s + sqrt(1 - (  (A)  ) ** 2)
  end
  (B) * 4
end
```

図 4: 円周率を計算するプログラム

問題 3

K 市の中心部は道が碁盤の目のように繋がっている．いま，南または東にのみ進んで 2 地点間を移動する経路の数を数えるプログラムを作りたい．

K 市の地図は以下のデータ構造として表す．各地点は，南北方向の座標 y と東西方向の座標 x で (y, x) と表す．ただし y および x は非負の整数であり，南ないし東に行くほど値は大きい．道は 2 次元配列 NS と WE で表す．$NS[y][x]$ は地点 (y, x) から南に向かう道がある場合に 1，そうでなければ 0 である．同様に，$WE[y][x]$ は地点 (y, x) から東に向かう道がある場合に 1，そうでなければ 0 である．これらの配列はプログラムのどこからでも参照できるとしてよい．図 5 に地図と NS と WE の対応の例を示す．

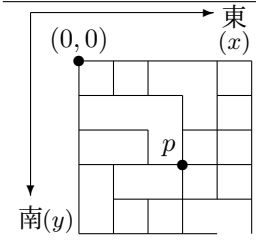
地図	NS	WE
	$\begin{bmatrix}[1,1,1,0,1,1],\\ [1,0,0,1,1,1],\\ [1,0,1,1,1,1],\\ [1,1,0,1,1,1],\\ [1,1,1,1,0,1],\\ [0,0,0,0,0,0]\end{bmatrix}$	$\begin{bmatrix}[1,1,1,1,1,0],\\ [1,1,1,0,1,0],\\ [1,1,0,1,1,0],\\ [1,1,1,1,1,0],\\ [0,1,1,1,1,0],\\ [1,1,1,1,0,0]\end{bmatrix}$

図 5: 地図と NS と WE の対応

以下の問に答えよ．

(1) 図 5 の地図において，地点 $(0, 0)$ から地点 $(3, 3)$（図中点 p）への経路は何通りあるか．

(2) 地点 $(0, 0)$ から地点 (y, x) への経路の数 $S(y, x)$ は以下の漸化式を満たす．空欄 (A) ～ (D) を NS と

WE を用いて埋めよ．

$$
\begin{aligned}
S(0,0) &= 1 \\
S(0,x) &= S(0,x-1) \times \boxed{\text{(A)}} && (x>0) \\
S(y,0) &= S(y-1,0) \times \boxed{\text{(B)}} && (y>0) \\
S(y,x) &= S(y,x-1) \times \boxed{\text{(C)}} + S(y-1,x) \times \boxed{\text{(D)}} && (x>0, y>0)
\end{aligned}
$$

(3) 設問 (2) の漸化式をふまえ，図 6 に示す再帰関数 cp1 を作った．空欄 (E) ～ (H) を埋めよ．

(4) cp1 を動的計画法を用いて改善し，図 7 のプログラム cp2 を作った．空欄 (I) ～ (N) を埋めよ．

(5) cp2 の計算量のオーダーを x と y を用いて表せ．導出の過程も簡単に示すこと．

```
def cp1(y, x)
  if y == 0
    if x == 0
      (E)
    else
      (F)
    end
  else
    if x == 0
      (G)
    else
      (H)
    end
  end
end
```

図 6: プログラム cp1

```
def cp2(y, x)
  count = make2d(y + 1, x + 1)
  for i in (I)
    for j in (J)
      count[i][j] = calc(i, j, count)
    end
  end
  count[y][x]
end
```

```
def calc(y, x, count)
  if y == 0
    if x == 0
      (K)
    else
      (L)
    end
  else
    if x == 0
      (M)
    else
      (N)
    end
  end
end
```

図 7: プログラム cp2