アルゴリズム入門　共通問題（2017年度Aセメスター試験）

[試験日時：2018年1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．

問題1

以下に示すロジスティック方程式は生物の個体数の変化を表すモデルのひとつである．

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = a \cdot x(t) - b \cdot x(t)^2$$

ここで $x(t)$ は時刻 t におけるその生物の個体数，a は自然増加率を表すパラメタ，b は餌や縄張りなどを巡る種内競争を表すパラメタである．微分 $\mathrm{d}x(t)/\mathrm{d}t$ の近似値として小さな正の値 h を用いた差分 $(x(t+h) - x(t))/h$ を用いることで，このモデルを計算機シミュレーションすることにした．なお初期値は $x(t) = x_0$（$t \leq 0$）とした．以下の問に答えよ．

(1) 差分で近似した後の方程式は，$x(t)$ を $x(t-h)$ から求める漸化式だとみなすことができる．変数 a，b，h を用いて空欄 [A]・[B] を埋め，この漸化式を完成させよ．

$$\begin{aligned} x(t) &= x_0 && \text{ただし } t \leq 0 \\ x(t) &= \boxed{\quad\text{A}\quad} \cdot x(t-h) - \boxed{\quad\text{B}\quad} \cdot x(t-h)^2 && \text{ただし } t > 0 \end{aligned}$$

(2) 問 (1) の漸化式をふまえ，$x(t)$ を再帰を用いて計算する関数 `population` およびループを用いて計算する関数 `population_loop` を作成した．空欄 [C] ～ [E] を埋めよ．2箇所の空欄 [D] には同じプログラム断片が入る．なお，引数 `t`，`a`，`b`，`h`，`x0` はそれぞれ漸化式の t，a，b，h，x_0 に対応し，整数ではなく実数が入力されるものとする．また，([E]).ceil() は [E] の値の小数点以下を切り上げた整数である．

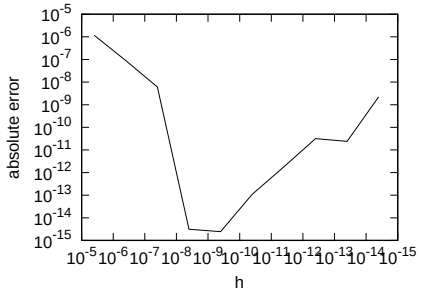
```
def population(t, a, b, h, x0)
  if [ C ]
    x0
  else
    x = population(t - h, a, b, h, x0)
    [ D ]
  end
end
```

```
def population_loop(t, a, b, h, x0)
  x = x0
  for i in 0 .. (([ E ]).ceil() - 1)
    x = [ D ]
  end
  x
end
```

(3) `population(0.4, 0.5, 0.1, 0.25, 1.0)` を計算する過程を再帰呼び出しの様子が分かるように説明せよ．計算過程が明確であれば計算結果を示す必要はない．

(4) `population` と `population_loop` を計算量の観点から比較せよ．

(5) 右のグラフの縦軸は `population` や `population_loop` で近似計算をした値と近似せず精密に計算した値との差の絶対値（誤差），横軸は近似計算に用いた h の値である．ある程度までは h が小さくなると誤差は減少するが，h がさらに小さくなると逆に誤差が大きくなった．この傾向は `population` と `population_loop` の両方で見られた．このような結果となった理由を述べよ．



問題 2

下方に示すプログラムに関連して，以下の問に答えよ．なお，rand() は 0 以上 1 未満の浮動小数点数を一様分布に従いランダムに返すものである．

(1) 辺の長さ 1 の正方形にランダムに点を n 個うち，ある頂点から距離 1 未満の点の数を数えることで四半円の面積を近似的に求める関数 circle(n) を定義した．[A]・[B] に入れる式を示せ．

(2) A 君は x と y の右辺が全く同じ式であることに気がついた．そこで共通の式をまとめて変数 r に代入しておき rand() の呼び出しを 1 回にまとめることを考えた．circle を書き換えて circle2 を定義したが，どうも四半円の面積ではないものを求めているようである．その理由と何を求めているかを述べよ．

(3) B 君は rand() が返すのが 0 から 1 の間の乱数なら，rand()**2 も 0 から 1 の間の数であることに気がついた．二乗すればよりランダムになるかと考え，circle を書き換えて circle3 を定義したが，どうも四半円の面積ではないものを求めているようである．その理由を述べよ．

(4) C 君は x と y の一つだけ新しい乱数を使い，もう一方は前回の乱数を再利用することを考えた．circle を書き換えて，circle4 を定義したところ，それらしい値を求めているようである．その理由を述べよ．

(5) x 軸が $[0, \pi)$ の範囲で，$\sin(x)$ 関数と x 軸で囲まれる範囲の面積を近似的に求める関数 wave(n) を定義した．[C] ～ [E] を埋めよ．なお，必要ならば円周率 PI や sin 関数 sin(x) などを利用しても良い．

```
def circle(n)
  count = 0.0
  for i in 0 .. (n - 1)
    x = rand()
    y = rand()
    if [ A ]
      count = count + 1
    end
  end
  [ B ]
end
```

```
def circle2(n)
  count = 0.0
  for i in 0 .. (n - 1)
    r = rand()
    x = r
    y = r
    if [ A ]
      count = count + 1
    end
  end
  [ B ]
end
```

```
def circle3(n)
  count = 0.0
  for i in 0 .. (n - 1)
    x = rand() ** 2
    y = rand() ** 2
    if [ A ]
      count = count + 1
    end
  end
  [ B ]
end
```

```
def circle4(n)
  count = 0.0
  y = rand()
  for i in 0 .. (n - 1)
    x = y
    y = rand()
    if [ A ]
      count = count + 1
    end
  end
  [ B ]
end
```

```
include(Math)
def wave(n)
  count = 0.0
  for i in 0 .. (n - 1)
    x = [ C ]
    y = rand()
    if [ D ]
      count = count + 1
    end
  end
  [ E ]
end
```

問題 3

$p_0, p_1, \ldots, p_{n-1}$ の n 人の交友関係を考える．各人の関係は二次元配列 rel に格納されており，p_i と p_j が直接の友人同士の場合 rel[i][j] および rel[j][i] の値は 1，そうでない場合には 0 である．また rel[i][i] の値は 0 である（自身自分は直接の友人ではない）．p_i と p_j，p_j と p_k が直接の友人同士のとき,「p_i と p_k は間に 1 人を介した友人」と言うことにする（p_i と p_k が直接の友人同士でも構わない）．間に 2 人を介した友人なども同様に定める．以上をふまえ以下の問に答えよ．なお，make1d(n) は長さ n で全要素が 0 の配列を作る関数であり，定義済みとする．

(1) rel が右のような配列であるとき，p_0, p_1, p_2, p_3 の 4 人のうち，間に 2 人を介した友人は誰と誰か．間にどの 2 人を介しているかもあわせて答えよ．

```
rel = [[0, 1, 1, 0],
       [1, 0, 0, 1],
       [1, 0, 0, 0],
       [0, 1, 0, 0]]
```

(2) p_i または p_j と直接の友人である人を全て挙げる関数 either を作成した．この関数は，i，j，n，rel を入力とし，長さ n の配列を返す．この配列の k 番目の要素は，p_k が p_i または p_j と友人であれば 1，そうでなければ 0 である．空欄 A ・ B を埋めよ．

```
def either(i, j, n, rel)
  union(rel[ A ], rel[ B ], n)
end
```

```
def union(r1, r2, n)
  r = make1d(n)
  for i in 0 .. (n - 1)
    if (r1[i] == 1 || r2[i] == 1)
      r[i] = 1
    end
  end
  r
end
```

(3) p_s と間に 1 人を介した友人を全て挙げる関数 friends1 を作成した．この関数は，s，n，rel を入力とし，長さ n の配列を返す．この配列の k 番目の要素は，p_k が間に 1 人を介した友人であれば 1，そうでなければ 0 である．空欄 C ～ E を埋めよ．

```
def friends1(s, n, rel)
  result = make1d(n)
  for i in 0 .. (n - 1)
    if rel[ C ][ D ] == 1
      result = union(result, [ E ], n)
    end
  end
  result
end
```

(4) p_s と間に k 人を介した友人を全て挙げる再帰関数 friends_k を作成した．この関数は，s，k，n，rel を入力とし，出力は friends1 と同様の形式である．空欄 F ～ H を埋めよ．

```
def friends_k(s, k, n, rel)
  if k == 0
    rel[ F ]
  else
    rel2 = friends_k(s, k - 1, n, rel)
    friends_sub(s, n, rel, rel2)
  end
end
```

```
def friends_sub(s, n, rel, rel2)
  result = make1d(n)
  for i in 0 .. (n - 1)
    if rel2[ G ] == 1
      result = union(result, [ H ], n)
    end
  end
  result
end
```

(5) friends_k(s,k,n,rel) の実行時間を n と k を用いてオーダ記法で示せ．その理由も簡単に説明すること．