

プログラムのページ

担当 森 口 繁 一

6303. 簡単なランダム・ウォークの実験

五十嵐 滋 (東大計数工学科)

X 軸上に束縛された動点が、時刻 $t=0$ では原点にあり、以後単位時間ごとにそれぞれ確率 $p, 1-p$ で $+1, -1$ だけ変位するものとする。次のプログラムの PART 1 はこのモデルをシミュレートして与えられた時間動点の位置を追跡し、毎時点での座標を印刷する。PART 2 はこの追跡を与えられたサンプル数だけ繰返し、最終の位置の度数及び相対度数を表にして印刷する。なお、このプログラムは東大計算センターの清水 留三郎氏の御好意により、同所の OKITAC 5090 のためのコンパイラでテストされたもので、ALGOLIP で記されている。コンパイルに要した時間はテープを掛けてから約 10 秒であった。

begin

real P, FLS;

integer CHI, LAMBDA, MU, NU, PP, TIME,
SAMPLE, X, T, S, MINX, MAXX, Z;

integer array FREQ [0 : 200];

procedure RANDOM;

comment

RANDOM assigns a new random integer to CHI, which is uniform between 0 and NU, and has NU as the period;

begin

CHI:=LAMBDA * CHI + MU;

CHI:=CHI - CHI + NU * NU

end RANDOM;

P:=READREAL;

TIME:=READINTEGER;

SAMPLE:=READINTEGER;

CHI:=READINTEGER;

LAMBDA:=READINTEGER;

MU:=READINTEGER;

NU:=READINTEGER;

PP:=FIX(P * FLOAT(NU));

PART 1:

X:=0;

for T:=1 step 1 until TIME do

begin

RANDOM;

X:=X+(if CHI≤PP then 1 else -1);

PRINTINTEGER (T);

PRINTINTEGER (X);

CRLF (1)

end;

PART 2:

MINX:=MAXX:=0;

for Z:=100-TIME step 2 until
100+TIME do

FREQ[Z]:=0;

for S:=1 step 1 until SAMPLE do

begin

X:=0;

for T:=1 step 1 until TIME do

begin

RANDOM;

X:=X+(if CHI≤PP then 1 else -1)

end;

FREQ[100+X]:=FREQ[100+X]+1;

if X<MINX then MINX:=X;

if X>MAXX then MAXX:=X

end;

CRLF (1);

PRINTREAL (P);

PRINTINTEGER (TIME);

PRINTINTEGER (SAMPLE);

CRLF (2);

for Z:=100+MINX step 2 until 100
+MAXX do

begin

PRINTINTEGER (Z-100);

PRINTINTEGER (FREQ[Z]);

PRINTINTEGER (FLOAT (FREQ[Z])/

FLS);

CRLF (1)

CRLF (1);
PRINTINTEGER (CHI)
end

データ・テープには乱数発生用の定数 CHI, LAM-BDA, MU, NU, 確率 $P(=p)$, 追跡時間 TIME, サンプル数 SAMPLE をこの順序で記す. 実施例のデータ・テープは

50000, 54321, 56789, 100000, 0.5, 20, 100,
である. $CHI \equiv 1 \pmod{20}$ であり, しかも MU は 2 または 5 を約数に持たないことから, この 5 桁の乱数の周期は 50000 であることが保証される.

6304. 三山くずしのプログラム

五十嵐 滋 (東大計数工学科)

“三山くずし” ([英] Nim) というのは基石, マッチ棒等を“石”, つまり構成要素, とする三つの“山”を作っておき, 2人の対局者が交互に任意の山を一つだけ選び, そこから1個以上の石を取り去って行き,

“順様式”のゲームでは最後の1石を取った方が勝, “逆様式”のゲームではそれを取らされた方が負と定められたゲームである. ここでは逆様式のゲームを計算機と人間とが対局する形でプログラムしてみた. その方式は次のとおりである.

(1) 新たなゲームを始める時(前のゲームを中断してもよい)にはデータ・テープに 0, 0, 0, とパンチして読み込ませる(キー・ボードから入れてもよい. 以下同じ)と, 準備をすていったん止る. 次に

(2) 三つの山の石の数を三つの整数としてテープから与える. ただし各山の石の数は100個以下とし, 山の順序は一定させておく.

(3) 計算機がこの三つの山を読み込むと, まずそれを1行にタイプする. もしゲームの最初に与えられた山ならば無条件で次に移るが, ゲームの途中の時は計算機側が最後に作った山の組み合わせと比較して人間が反則しなかったかどうか確かめ, 反則があったなら ‘DO NOT REVOKE, PLEASE.’ とタイプしていったん止る. その時は人間はやりなおした手を(2)に従って再び読み込めばよい. こうして反則のない組み合わせを読んだら次に移る.

(4) 後述の戦略とルールとに従って計算機は一つの山を選び何個か石を減らし, その結果できた山の組み合わせを1行にタイプする. いったん止って人間の手を待つ.

(5) 人間はタイプされた結果を見て, 同様に一つの山を選び石を減らして, 新しい組み合わせを作り再び(2)に戻って計算機に読み込ませる. 勝負がつくまでは以上のステップが繰り返される.

(6) データ・テープに 0, 0, 1 または 0, 1, 0 または 1, 0, 0 の組が(反則なく)与えられた時は計算機の負なので, データの復誦印刷はせずに ‘YOU HAVE WON.’ とタイプして止まる. 逆の場合はまず ‘I HAVE WON.’ とタイプしたのち, 組み合わせを次の行にタイプして止る.

(戦略) 次の戦略はよく知られている. (cf. 森口繁一監訳(1960), 『システム工学』, JUSE 出版社 etc.) 与えられた三つの山の石の数を N_i ($i=1, 2, 3$) それを2進数になおしたものの 2^j の位を B_{ij} とする. また N_i の内0でないものの数を C , 丁度1であるものの数を $C1$ で表わす. (i) $C-C1 \geq 2$ のとき, $S_j = \sum_i B_{ij}$ を作る. (a) S_j がことごとく偶数でないとき, そのような j のうち最大のものを k とおき, $B_{ik} = 1$ である i が存在するからその一つを取って i_0 とする. $M_i = N_i$ ($i \neq i_0$), $0 \leq M_{i_0} < N_{i_0}$ を満たす数の組 M_1, M_2, M_3 について上と同様の S_j を作ったとき, S_j がことごとく偶数となるような M_{i_0} がちょうど一つ存在する. この組を計算機の手とする. (b) S_j がことごとく偶数のとき, 計算機は負ける運命にあるわけだが, 人間が途中で計算違いをしてくれれば(a)の状態になるので, たとえば最大の N_i を N_{i_0} としたとき $M_{i_0} = N_{i_0} - 1$ として手とする. (ii) $C-C1 = 1$ のとき, N_i の内で2以上のはただ一つであるからこれを N_{i_0} とする. $C1$ の偶奇に応じて M_{i_0} を 1, 0 にすれば場には奇数個の孤立した石が残るので, 計算機の勝は明らかとなる. (iii) $C-C1 = 0$ のとき, $C1$ 個の孤立した石が与えられたのでその偶奇により勝負は定まっている. ((i) (a) の状態の次に再び計算機の手番になったとき必ず (i) (a) または (ii) の状態が起ることは明らかである. だからいったん (i) (a) の状態を経過してこの戦略を続けていくと必ず (ii) の状態が起り計算機が勝つことになる. なお最初の山をランダムに作ると (a) になる割合は (b) になる割合よりずっと多いので後手すなわち計算機が有利である) プログラム中 D は人間の触れなかった山の数, $D1$ は触れた山の数で反則の検出に使っている. また $L2:U:=3$ の所と $L3$ とを除き数字3を全部数字 n に変えれば “ n 山くずし” のプログラムとなり, PARITY の部分他を少し変えると “合計が予じめ規