HW7 - Do not Post

2) [A] The following linear homogeneous recurrence:

$$x_N(i) = \frac{1}{2}(x_N(i-1) + x_N(i+1))$$

is true because the game is played with a fair coin, meaning that there is a 0.5 chance of gaining a dollar ($x_N(i+1)$) or losing a dollar ($x_N(i-1)$). The probabilities are added because they are independent events. The boundary conditions

$$x_N(0)=0, \quad x_N(N)=1$$

are true because if you enter the game with nothing, you have nothing to bet, therefore you never leave a winner. If you enter the game with the max capital, then you've won, so you will always exit the game as a winner

[B] $x_N(0)=0$ $\quad$ $x_N(N)=1$

$y_N(0) = \frac{0}{N} = 0$ ✓ $\quad$ $y_N(N) = \frac{N}{N} = 1$ ✓

Suppose $x_N(N) = \frac{i}{N}$

$y_N(i) = \frac{i}{N} = x_N(N) = \frac{1}{2}(x_N(i-1) + x_N(i+1))$

$\frac{i}{N} = \frac{1}{2}\left(\frac{i-1}{N} + \frac{i+1}{N}\right)$

$\frac{i}{N} = \frac{1}{2}\left(\frac{2i}{N}\right)$

$\frac{i}{N} = \frac{i}{N}$ ✓

The explicit formula $\frac{i}{N}$ exists in this situation because there is an equal probability of losing or gaining a dollar because of the fair coin c The linear inhomogeneous recurrence:

$$E_N(i) = \frac{1}{2}\left[E_N(i-1) + E_N(i+1)\right] + 1$$

is an accurate representation of the expected number of rounds to complete a Gambler's Run game because, like in the previous problem, with the fair coin there is an equal chance of gaining a dollar and getting to the max capital or losing a dollar and leaving a loser. Again, these are independent probabilities, so they're added together. The addition of the 1 at the end counts entering the casino to bet.

D $E_N(0) = 0(N-0) = 0$ ✓

$E_N(N) = N(N-N) = 0$ ✓

* Suppose $E_N = i(N-i)$

$$i(N-i) = \frac{1}{2}(E_N(i-1) + E_N(i+1)) + 1$$

$$i(N-i) = \frac{1}{2}((i-1)(N-i+1) + (i+1)(N-i-1)) + 1$$

$$i(N-i) = \frac{1}{2}(Ni - i^2 + i - N + i - 1 + Ni - i^2 - i + N - i - 1) + 1$$

$$i(N-i) = \frac{1}{2}(2Ni - 2i^2 - 2) + 1$$

$$i(N-i) = Ni - i^2 - 1 + 1$$

$$i(N-i) = i(N-i)$$ ✓

This explicit formula $i(N-i)$ exists to characterize the average number of rounds to exit Gambler's Run only because the probability of gaining enough to win or losing all the money is exactly the same.

```
> #Do not post
#Nikita John, September 27th, 2021, Assignment 7
 #Maple Code for Dr. Z.'s Dynamcial Methods in Biology, Lecture 7

Help7 :=proc( ) :
      print( ` GR(p,i,N), GRt(p,i,N),  GRm(N,p), OneStepMarkov(P,i), MarkovTrip(P,K), StSa(P,K)
     ,  StS(P), StSp(P,K), RandSM(N)   `) : end:

with(Statistics) :
with(LinearAlgebra) :


     #GR(p,i,N): Simulating a Gambler's Ruin problem. Ypu enter a casino with i dollars. At every
     round you win a dollar with prob. p and lose a

     #dollar with prob. 1-p. You exit as soon as you are broke or got the maximum allowed  that is
     N. It  returbs [0,c] or [1,c]
#if you lost or won respectively, where c is the number of rounds. Try: GR(3/5,5,10);
GR :=proc(p, i, N) local X, x, d, c :
X := RandomVariable(Bernoulli(p)) :
c := 0 :
x := i :
 print( `I enter the casino with `, i, `dollars `) :
while x > 0 and x < N do
 c := c + 1 :
 d := trunc(Sample(X, 1)[1]) :
  if d = 0 then
    x := x-1 :
   print( `I lost a dollar, now I have`, x, `dollars `) :
   else
  x := x + 1 :
   print( `I lost a dollar, now I have`, x, `dollars `) :
  fi:
 od:
 if x = 0 then
  print( `I am so sad, I am broke`) :
  print( `It took`, c, `rounds `) :
   RETURN([0, c]) :
  elif x = N then
   print( `Yea, I am rich, I now have the max`, N) :
    print( `It took`, c, `rounds `) :
   RETURN([1, c]) :
  fi:
end:


#GRt(p,i,N): A terse version of GR(p,i,N). Returns [0,c] or [1,c], where c is the number of rounds
GRt :=proc(p, i, N) local X, x, d, c :
X := RandomVariable(Bernoulli(p)) :
```

```
c := 0 :
x := i :
while x > 0 and x < N do
 c := c + 1 :
 d := trunc(Sample(X, 1)[1]) :
  if d = 0 then
   x := x-1 :
   else
  x := x + 1 :
  fi:
 od:
 if x = 0 then
   RETURN([0, c]) :
  elif x = N then
   RETURN([1, c]) :
  fi:
end:


#GRm(N,p): The (N+1)x(N+1) Transition matrix for a Gambler's ruin with max. capital N
GRm :=proc(N, p) local i :
[
[1, 0$N],
seq([0$(i-1), (1-p), 0, p, 0$(N-1-i)], i = 1 ..N-1),
[0$N, 1]
 ]:
end:



    ##OneStepMarkov(P,i): Suppose that there are n sites, let P be an n by n  transition matrix
    given as a list of lentgh n
#where P[i][j] is the probability of getting from site i to site j. Outputs the state visited next

OneStepMarkov :=proc(P, i)
trunc(Sample(RandomVariable(ProbabilityTable(P[i])), 1)[1]) :
end:

#MarkovTrip(P,K): The trajectory of a Markov chain with transition matrix P with K steps
#Try: MakovTrip([[0.2,0.8],[0.4,0.6]],1000);
MarkovTrip :=proc(P, K) local x, L, i :

x := 1 :
L := [1] :

for i from 1 to K do
x := OneStepMarkov(P, x) :
L := [op(L), x] :
od:
```

```
L :

end:


    #StSa(P,K): Given a transition matrix, ESTIMATES the steady-state probability distribution
    of the Markov chain using  a walk with K steps
StSa :=proc(P, K) local L, x, f, i :
L := MarkovTrip(P, K) :
f := add(x[L[i]], i = 1 ..K) :
evalf ( [seq(coeff ( f, x[i], 1) / K, i = 1 ..nops(P) ) ]) :

end:


    #StS(P,K): Given a transition matrix, FIND  the EXACT steady-state probability distribution
    of the Markov chain using  linear algebra
StS :=proc(P) local x, n, eq, var, i, j :
n := nops(P) :
var := {seq(x[i], i = 1 ..n) } :
eq := {seq( add(x[i] * P[i][j], i = 1 ..n) = x[j], j = 1 ..n), add(x[i], i = 1 ..n) = 1 } :
subs(solve(eq, var), [seq(x[i], i = 1 ..n) ]) :
end:


    #StSp(P,K): Given a transition matrix, FIND  the approximate steady-state probability
    distribution  using raising the matrix to  the K-th power
StSp :=proc(P, K) local Q, i :
Q := evalm(P&^K) :
[seq(Q[i, 1], i = 1 ..nops(P) ) ] :
end:


#RandPV(N): A radnom stochastc  vector of length N
RandPV :=proc(N) local ra, i, v, s :
ra := rand(1 ..1000) :
v := [seq(ra( ), i = 1 ..N) ] :
s := add(v[i], i = 1 ..N) :
v / s :
end:

#Added after class
#RandSM(N): A random N by N stochastic matrix
RandSM :=proc(N) local i :
[seq(RandPV(N), i = 1 ..N) ] :
```

```
end:

> #1: EstGR computes GRt K times and gives win probability and average rounds it takes
EstGR := proc(p, i, N, K) local X, x, d, c, Win, Loss, roundNum, inc, storeC :
X := RandomVariable(Bernoulli(p)) :
c := 0 :
x := i :
inc := 1;
Win := 0 :
Loss := 0 :
storeC := 0 :
while inc ≤ K do
while x > 0 and x < N do
c := c + 1 :
d := trunc(Sample(X, 1)[1]) :
 if d = 0 then
  x := x-1 :
 else
 x := x + 1 :
 fi:
od:
if x = 0 then
  Loss := Loss + 1 :
 elif x = N then
  Win := Win + 1 :
 fi:
if inc = 1 then
storeC := c :
else
storeC := storeC + c :
fi:
inc := inc + 1 :
od:
RETURN([Win/K, storeC/K]) :
end:

> #3: Coding ExactFair
ExactFairGR := proc(i, N) local X, c :
X := i/N :
c := i·(N − i) :
RETURN([X, c]) :
end:

> #i = 1

    #Note: I inferred that ExactFairGR and EstGR should yield the same answer, however for
    some reason it wasnt working. I assume that its a problem with my EstGR code, however
    formulaically it makes sense to me so I'm not sure how to improve it. Any suggestions would
    be greatly appreciated!
```

$ExactFairGR(1, 20);$

$EstGR\left(\frac{1}{2}, 1, 20, 3000\right);$

$$\left[\frac{1}{20}, 19\right]$$
$$[0, 1] \quad (1)$$

> $\#i = 2$

$ExactFairGR(2, 20);$

$EstGR\left(\frac{1}{2}, 2, 20, 3000\right);$

$$\left[\frac{1}{10}, 36\right]$$
$$[0, 4] \quad (2)$$

> $\#i = 3$

$ExactFairGR(3, 20);$

$EstGR\left(\frac{1}{2}, 3, 20, 3000\right);$

$$\left[\frac{3}{20}, 51\right]$$
$$[1, 71] \quad (3)$$

> $\#i = 4$

$ExactFairGR(4, 20);$

$EstGR\left(\frac{1}{2}, 4, 20, 3000\right);$

$$\left[\frac{1}{5}, 64\right]$$
$$[0, 108] \quad (4)$$

> $\#i = 5$

$ExactFairGR(5, 20);$

$EstGR\left(\frac{1}{2}, 5, 20, 3000\right);$

$$\left[\frac{1}{4}, 75\right]$$
$$[0, 49] \quad (5)$$

> $\#i = 6$

$ExactFairGR(6, 20);$

$EstGR\left(\frac{1}{2}, 6, 20, 3000\right);$

$$\left[\frac{3}{10}, 84\right]$$
$$[0, 72] \quad (6)$$

```
> #i = 7
ExactFairGR(7, 20);
EstGR(1/2, 7, 20, 3000);
```

$$\left[\frac{7}{20}, 91\right]$$
$$[0, 29] \tag{7}$$

```
> #i = 8
ExactFairGR(8, 20);
EstGR(1/2, 8, 20, 3000);
```

$$\left[\frac{2}{5}, 96\right]$$
$$[1, 138] \tag{8}$$

```
> #i = 9
ExactFairGR(9, 20);
EstGR(1/2, 9, 20, 3000);
```

$$\left[\frac{9}{20}, 99\right]$$
$$[1, 169] \tag{9}$$

```
> #i = 10
ExactFairGR(10, 20);
EstGR(1/2, 10, 20, 3000);
```

$$\left[\frac{1}{2}, 100\right]$$
$$[0, 44] \tag{10}$$

```
> #i = 11
ExactFairGR(11, 20);
EstGR(1/2, 11, 20, 3000);
```

$$\left[\frac{11}{20}, 99\right]$$
$$[0, 29] \tag{11}$$

```
> #i = 12
ExactFairGR(12, 20);
EstGR(1/2, 12, 20, 3000);
```

$$\left[\frac{3}{5}, 96\right]$$

$$[0, 36] \quad (12)$$

```
> #i = 13
ExactFairGR(13, 20);
EstGR(1/2, 13, 20, 3000);
```

$$\left[\frac{13}{20}, 91\right]$$
$$[0, 35] \quad (13)$$

```
> #i = 14
ExactFairGR(14, 20);
EstGR(1/2, 14, 20, 3000);
```

$$\left[\frac{7}{10}, 84\right]$$
$$[1, 26] \quad (14)$$

```
> #i = 15
ExactFairGR(15, 20);
EstGR(1/2, 15, 20, 3000);
```

$$\left[\frac{3}{4}, 75\right]$$
$$[0, 21] \quad (15)$$

```
> #i = 16
ExactFairGR(16, 20);
EstGR(1/2, 16, 20, 3000);
```

$$\left[\frac{4}{5}, 64\right]$$
$$[1, 158] \quad (16)$$

```
> #i = 17
ExactFairGR(17, 20);
EstGR(1/2, 17, 20, 3000);
```

$$\left[\frac{17}{20}, 51\right] \quad (17)$$

```
> #i = 18
ExactFairGR(18, 20);
EstGR(1/2, 18, 20, 3000);
```

$$\left[\frac{9}{10}, 36\right]$$

$$[1, 22] \quad (18)$$

> *#i = 19*
> *ExactFairGR*(19, 20);
> *EstGR*$\left(\frac{1}{2}, 19, 20, 3000\right)$;

$$\left[\frac{19}{20}, 19\right]$$
$$[1, 1] \quad (19)$$

> *#4*
> *A* := *RandSM*(10);

$$A := \left[\left[\frac{20}{921}, \frac{110}{921}, \frac{197}{4605}, \frac{283}{4605}, \frac{352}{4605}, \frac{17}{4605}, \frac{248}{1535}, \frac{247}{1535}, \frac{286}{1535}, \frac{763}{4605}\right], \left[\frac{81}{1877}, \frac{137}{1877}, \frac{90}{1877}, \frac{156}{1877}, \frac{171}{3754}, \frac{595}{3754}, \frac{25}{3754}, \frac{301}{1877}, \frac{605}{3754}, \frac{414}{1877}\right], \left[\frac{477}{3356}, \frac{199}{1678}, \frac{25}{1678}, \frac{379}{3356}, \frac{175}{3356}, \frac{2}{839}, \frac{181}{1678}, \frac{87}{1678}, \frac{235}{839}, \frac{393}{3356}\right], \left[\frac{845}{4964}, \frac{11}{292}, \frac{164}{1241}, \frac{897}{4964}, \frac{227}{2482}, \frac{77}{4964}, \frac{311}{2482}, \frac{39}{4964}, \frac{151}{1241}, \frac{583}{4964}\right], \left[\frac{41}{342}, \frac{359}{1881}, \frac{509}{3762}, \frac{239}{3762}, \frac{367}{3762}, \frac{17}{1254}, \frac{565}{3762}, \frac{2}{19}, \frac{1}{198}, \frac{149}{1254}\right], \left[\frac{117}{691}, \frac{173}{2764}, \frac{643}{5528}, \frac{817}{5528}, \frac{177}{5528}, \frac{707}{5528}, \frac{869}{5528}, \frac{357}{5528}, \frac{101}{2764}, \frac{237}{2764}\right], \left[\frac{458}{4411}, \frac{155}{4411}, \frac{482}{4411}, \frac{584}{4411}, \frac{628}{4411}, \frac{751}{4411}, \frac{130}{4411}, \frac{585}{4411}, \frac{31}{401}, \frac{27}{401}\right], \left[\frac{21}{409}, \frac{772}{4499}, \frac{172}{4499}, \frac{71}{409}, \frac{147}{4499}, \frac{266}{4499}, \frac{13}{409}, \frac{64}{409}, \frac{521}{4499}, \frac{762}{4499}\right], \left[\frac{260}{1731}, \frac{85}{577}, \frac{308}{5193}, \frac{256}{5193}, \frac{409}{5193}, \frac{253}{1731}, \frac{280}{1731}, \frac{361}{5193}, \frac{625}{5193}, \frac{10}{577}\right], \left[\frac{786}{3289}, \frac{683}{3289}, \frac{372}{3289}, \frac{183}{3289}, \frac{109}{3289}, \frac{40}{3289}, \frac{17}{3289}, \frac{454}{3289}, \frac{28}{299}, \frac{337}{3289}\right]\right] \quad (20)$$

> *evalf*(*StSa*(*A*, 4000));

[0.1182500000, 0.1100000000, 0.07825000000, 0.09950000000, 0.06700000000, 0.07175000000, 0.1032500000, 0.1110000000, 0.1257500000, 0.1152500000] **(21)**

> *evalf*(*StS*(*A*));

[0.1188198785, 0.1182894109, 0.07789975470, 0.1030479142, 0.06686855912, 0.07244091436, 0.08954665615, 0.1087605099, 0.1242610846, 0.1200653176] **(22)**

> *evalf*(*StSp*(*A*, 4000));

[0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785, 0.1188198785] **(23)**

>