The Longest Run Of Heads

Constantin Ioana Teodora + Ion Melania Victorita

grupa 241

Introducere

Acest proiect analizeaza teoretic si prezinta intr-un mod interactiv fenomenul pe cat de intrigant, pe atat de abordat de diverse persoane pasionate de acest domeniu al probabilitatilor. Este vorba despre problema "The Longest Run Of Heads", pe baza careia profesorul Mark F. Schilling a redactat o lucrare revelatoare.

Acesta a pus urmatoarea problema :

The two sequences shown below each purportedly represent the results of 200 tosses of a fair coin. One of these is an actual sequence obtained from coin tossing, while the other sequence is artificial. Can you decide, in sixty seconds or less, which of the sequences is more likely to have arisen from actual coin tossing and which one is the imposter?

Sequence #1

T HHHHT T T T HHHHT HHHHHHHHHT T THHT T HHHHHT T T T T T HHT HHT HHH T
T T HT T HHHHT HT T T HT T T HHT T T T HHHHHHT T T HHT T HHHT HHHHHHT T T T
T HT T T HHT T HT T HHT T T HHT T T HH THHT HHT T T T T HHT HHHHHHT HT HT T
HT HT T HHHT T HHT HT HHHHHHHHT THT T HHHT HHT T HT T T T T T HHHT HHH

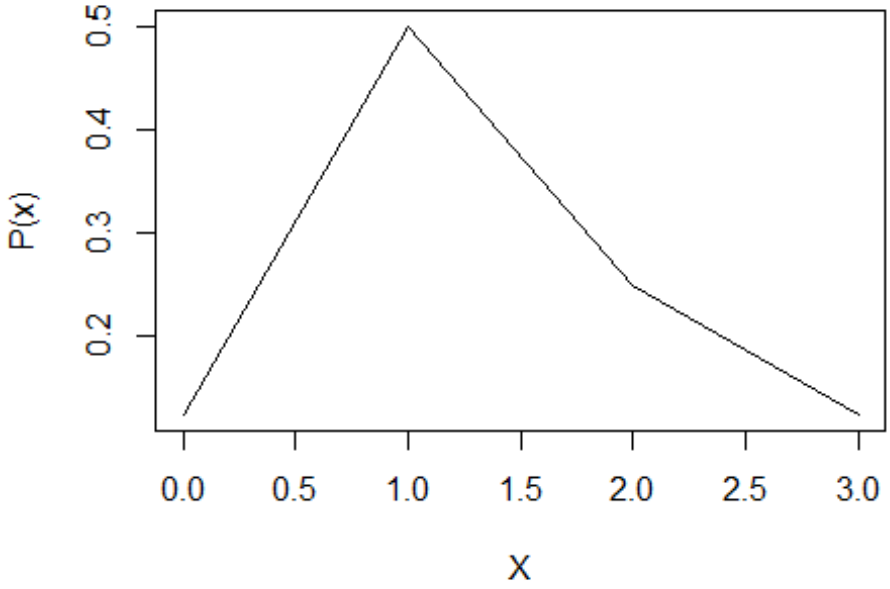
Sequence #2

T HT HT T T HT T T T T HT HT T T HT T HHHT HHT HT HT HT T T T HHT T HHT T HHH T
HHHT T HHHT T T HHHT HHHHT T T HT HT HHHHT HT T T HHHT HHT HT T T HHT H
HHT HHHHT T HT HHT HHHT T T HT HHHT HHT T T HHHT T T T HHHT HT HHHHT H
T,T HHT T T T HT HT HT T HT HHT T HT T THT T T T HHHHT HT HHHT T HHHHHT HH

El a pornit de la urmatoarea distributie a unei secvente de aruncari. A analizat, pentru inceput cazul in care sunt efectuate 3 aruncari. Obtinand urmatoarele posibile secvente: HHH, HHT, HTH, HTT, THH, THT, TTH, TTT

Pe baza acestor date, poate fi efectuata urmatoarea distributie:

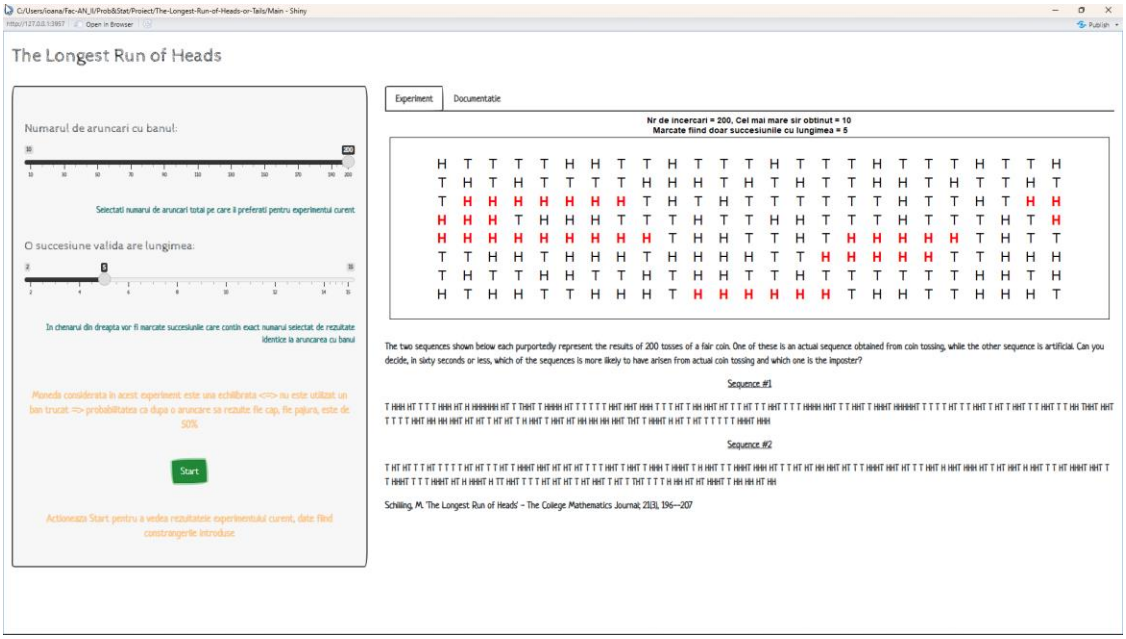
longest head run	*probability*
0	1/8
1	4/8
2	2/8
3	1/8



Astfel, in cazul unei secvente de 3 aruncari, probabilitatea ca cea mai mare secventa sa fie egala cu x este P(x).

De la acest experiment incipent, problema in cauza a fost derivata de profesorul de la California State University si a fost exepmplificata prin intermediul unei aplicatii ce foloseste functionalitatile pachetului Shiny de catre noi, autorii lucrarii curente.

Descrierea aplicatiei



Ce are ea in componenta?

1. Fisierul “app.R”, care inglobeaza interfata cu care un user al programului intra in contact

Aceasta ii alipeste aplicatiei o tema atractiva, ii ofera un layout, alcatuit dintr-un side-panel cu care userul interactioneaza si un panou principal impartit in numeroase tab-uri, unde acesta vede rezultatul testului cerut.

Si, de asemenea, tot in acest fisier este inclusa si comunicarea cu serverul, oferind astfel un output, n rezultat pentru datele introduce de user.

Codul sursa:

```
library(shiny)
library(bslib)

# Define UI for application that draws a histogram
ui <- fluidPage(

# stabilirea unei teme pentru ui
theme = bs_theme(
  version = 4,
  bootswatch = "sketchy"
),

tags$head(tags$link(rel = "icon", type = "image/x-icon",
                    href = "https://o.remove.bg/downloads/ff8780c1-3a97-4150-
9093-822aa4576925/pngtree-vector-coins-icon-dollar-gold-coin-png-
```

```
image_2462733-removebg-preview.png")),

tags$title(class="text-center bg-succes text-muted", "Longest run of heads"),


# Application title
titlePanel("The Longest Run of Heads"),
br(),


# Sidebar with a slider input for number of bins
sidebarLayout(

    sidebarPanel(
        br(),
        sliderInput("trials",
                    label = h5("Numarul de aruncari cu banul:"),
                    min = 10,
                    max = 200,
                    step = 10,
                    value = 200), br(),

        div("Selectati numarul de aruncari total pe care il preferati pentru
experimentul curent",
            style = "font-size: 11pt;color:teal",align="right"), br(),

        sliderInput("minlength",
                    label = h5("O succesiune valida are lungimea:"),
                    min = 2,
                    max = 15,
                    step = 1,
                    value = 5), br(),


        div("In chenarul din dreapta vor fi marcate succesiunile care contin
exact numarul selectat de rezultate identice la aruncarea cu banul",
        style = "font-size: 11pt;color:teal",align="right"), br(),

    br(), br(),
    div("Moneda considerata in acest experiment este una echilibrata <=> nu
este utilizat un ban trucat => probabilitatea ca dupa o aruncare sa rezulte
fie cap, fie pajura, este de 50%",
        style = "font-size: 13pt;color:#fcc780",align="center"), br(),

    br(),
    div(actionButton("start", label="Start", class = "btn-success"),
align="center"), br(), br(),
    div("Actioneaza", tags$b("Start")," pentru a vedea rezultatele
experimentului curent, date fiind constrangerile introduse", style = "font-
```

```
size: 13pt;color:#fcc780", align="center"),br(),



    ),
    # Show a plot of the generated distribution
    mainPanel(

      tabsetPanel(type = "tab",
                  tabPanel("Experiment", plotOutput("experiment"),
                  p("The two sequences shown below each purportedly represent
the results of 200 tosses
    of a fair coin. One of these is an actual sequence obtained from coin
tossing, while
    the other sequence is artificial. Can you decide, in sixty seconds or
less, which of
    the sequences is more likely to have arisen from actual coin tossing and
which one is
    the imposter?" ,style="margin-top:-10px"),
                            p("Sequence #1", style = "text-align: center;text-
decoration: underline"),
                            p("T HHH HT T T T HHH HT H HHHHHH HT T THHT T HHHH
HT T T T T T HHT HHT HHH T T T HT T HH HHT HT T T HT T T HHT T T T HHHH HHT T
T HHT T HHHT HHHHHT T T T T HT T T HHT T HT T HHT T T HHT T T HH THHT HHT T T
T T HHT HH HH HHT HT HT T HT HT T H HHT T HHT HT HH HH HH HHT THT T HHHT H HT
T HT T T T T T HHHT HHH"),
                            p("Sequence #2", style = "text-align: center;text-
decoration: underline;"),
                            p("T HT HT T T HT T T T T HT HT T T HT T HHHT HHT
HT HT HT T T T HHT T HHT T HHH T HHHT T H HHT T T HHHT HHH HT T T HT HT HH
HHT HT T T HHHT HHT HT T T HHT H HHT HHH HT T HT HHT H HHT T T HT HHHT HHT T
T HHHT T T T HHHT HT H HHHT H TT HHT T T T HT HT HT T HT HHT T HT T THT T T T
H HH HT HT HHHT T HH HH HT HH"),
                            p("Schilling, M. 'The Longest Run of Heads' - The
College Mathematics Journal; 21(3), 196--207")),

                  tabPanel("Documentatie", htmlOutput("documentatie")),
                  tabPanel("Documentatie", htmlOutput("documentatie"),
                           p("De-a dreptul contraintuitiv, raspunsul la
dilema anterioara, si anume 'Care este secventa constituita din date
adevarate?', il reprezinta primul set de aruncari cu banul, cel care are cea
mai lunga succesiune de rezultate
identice (si anume 8), in timp ce prima detine un record de numai 5. Inainte
de a citi mai departe, va sugeram sa
alocati putin timp de gandire urmatoarei intrebari: Puteti găsi
probabilitatea ca cea mai lungă secventa consecutiva
de 'Heads' sa aiba lungimea 'x' din 'n' aruncări a unei monede corecte?"),

                            p("Fiecare astfel de probabilitate poate fi gasita
prin partitionarea evenimentelor care implica secvente mai scurte,
```

conditionand modul de incepere al secventelor. Asadar, probabilitatea dorita va fi suma acestor probabilitati ale
  evenimentelor in care avem secvente mai scurte. Astfel, putem gasi raspunsul la intrebare plecand de la aceste cazuri si
construind solutia pas cu pas. In acest caz, ar fi mai simplu să consideram evenimentele unde cele mai lungi secvente
au o lungime mai mică sau egală cu o anumită sumă decât exact egala cu un numar."),

                            p("Astfel, sa luam un exemplu:"),
                            p("Evenimentul pentru care dorim să găsim probabilitatea poate fi definit ca:"),
                            p(" B = {cea mai lunga secventa de 'heads' (capete) este egala cu 3 in 10 aruncari}."),

                            p("Vom defini in continuare:"),
                            p("R3 = {cea mai lunga secvente de capete este mai mica sau egala cu 3} = {X ≤ 3}"),
                            p("R2 = {cea mai lunga secvente de capete este mai mica sau egala cu 2} = {X ≤ 2},
unde X reprezinta lungimea celei mai lungi secvente de capete consecutive în cele 10 aruncari.
Putem nota ca R2 ⊂ R3 and B = R3 ∩ R2c de unde rezulta ca avem exact 3. Astfel,"),
                            p("P (B) = P (R3) – P (R2)"),
                            p("cum B contine toate elementele din R3 care nu sunt in R2. Urmatorul pas ar fi sa gasim
P(R3). Elementele din R3 pot fi impartite in functie de cum incep secventele:"),
                            p("R30 = {secventele care incep cu T unde X ≤ 3}"),
                            p("R31 = {secventele care incep cu HT unde X ≤ 3}"),
                            p("R32 = {secventele care incep cu HHT unde X ≤ 3}"),
                            p("R33 = {secventele care incep cu HHHT unde X ≤ 3}"),
                            p("Cu alte cuvinte ,secventele din R3 incep cu 0,1,2 sau 3 capete. De asemenea, submultimile sun disjuncte, ceea ce conduce la urmatorul aspect:"),
                            p("R3 = R30 ∪ R31 ∪ R32 ∪ R33, si"),
                            p("P (R3) = P (R30) + P (R31) + P (R32) + P (R33) "),
                            p("Pentru a gasi toate probabilitatile, continuam cu urmatoarea intrebare despre cate secvente exista din 'n' aruncari cu moneda
 in care cea mai lunga secventa de capete este mai mica sau egala cu 'x'. Vom nota astfel An(x) si observam ca
P (R3) = A10(3)/2^10 si A10(3) = A9(3) + A8(3) + A7(3) + A6(3) , din moment ce P (R30(3)) = A9(3)/2^10 si asa mai departe.

```
Pentru n<=3 toate secventele vor avea cea mai mare lungime de capete mai mica
sau egala cu 3. Vom defini astfel:"),
                          p("An(3) = 2^n, pentru  n = 0, 1, 2, 3"),
                          p("An(3) = An–1(3) + An–2(3) + An–3(3) + An–4(3) ,
pentru  n = 4, 5, . . ."),
                          p("Punand toate acestea impreuna, probabilitatea
finala va fi: "),
                          p("P (B) = P (R3) – P (R2) = (773 / 1024) - (504 /
1024) = 269/1024 = 0.263")),
      )
    )
  )
)


source("main.R")

options(shiny.error = browser)

server <- function(input, output) {

dataInput <- reactive({
  tosses.gen(input$trials*(input$start>-1), 0.5)
})

output$experiment <- renderPlot({
  plot.gen(input$minlength, dataInput())
})

}

# Run the application
shinyApp(ui = ui, server = server)
```

2. Un fisier “main.R”, ce reprezinta functiile care, dat fiind inputul introdus de catre user, analizeaza aceste date si le folosesc pentru a calcula rezultatul final al aplicatiei

Fisierul contine o functie care simuleaza „datul cu banul”, considerand o moneda echilibrata (tosses.gen), si o functie care deseneaza graficul corespunzator acestui set de date, marcand totodata secventele de lungime maxima generate (plot.gen).

Codul sursa:

```
tosses.gen <- function(trials,prob){

# generate values from a uniform distribution
tosses <-runif(trials) #trials = number of observations desired

# the length of the vector "faces" is equal to the number of tosses
# will represent the result obtained with each throw (H or T)
```

```
faces <- seq(1,length(tosses))

# when we toss the coin we can get either heads which we will mark with 1
# or tails which we will mark with 0
for (i in 1:length(tosses)){
  if (tosses[i]<=prob){
    faces[i] = 1}
  else faces[i] = 0
}

# we will use a "data.frame" type structure, initialized with NA,
# to store our future data
mydata <- data.frame(matrix(data = NA,ncol=3,nrow=(length(faces))))
colnames(mydata) <- c("face","id","longest_run")

#the first column will represent the result of each coin toss (1 or 0)
mydata[,1]<-faces

# we will use an ID number for each run
# therefore that is what our second column will store
id <- 1
mydata[1,2] <- 1  # 1 is the start number (id)

for (i in 2:length(faces)){

  f1 <- mydata[i-1,1]  #the previous result
  f2 <- mydata[i,1]    #the current result we are looking at

  #if they have the same value, they will receive the same id
  if (f1 == f2) mydata[i,2] <- id
  else {
    id <- id +1
    mydata[i,2] <- id}
}

# all_Runs will store the observed frequency of each id value,
# determining the exact run length for each run
all_Runs <- table(mydata[,2])

# the third column will represent the corresponding run length
# for each unique value of id
val<-2
curr<-1
mydata[1,3] <- all_Runs[curr]

while(val <= nrow(mydata)){
  i1 <- mydata[val-1,2]
  i2 <- mydata[val,2]
```

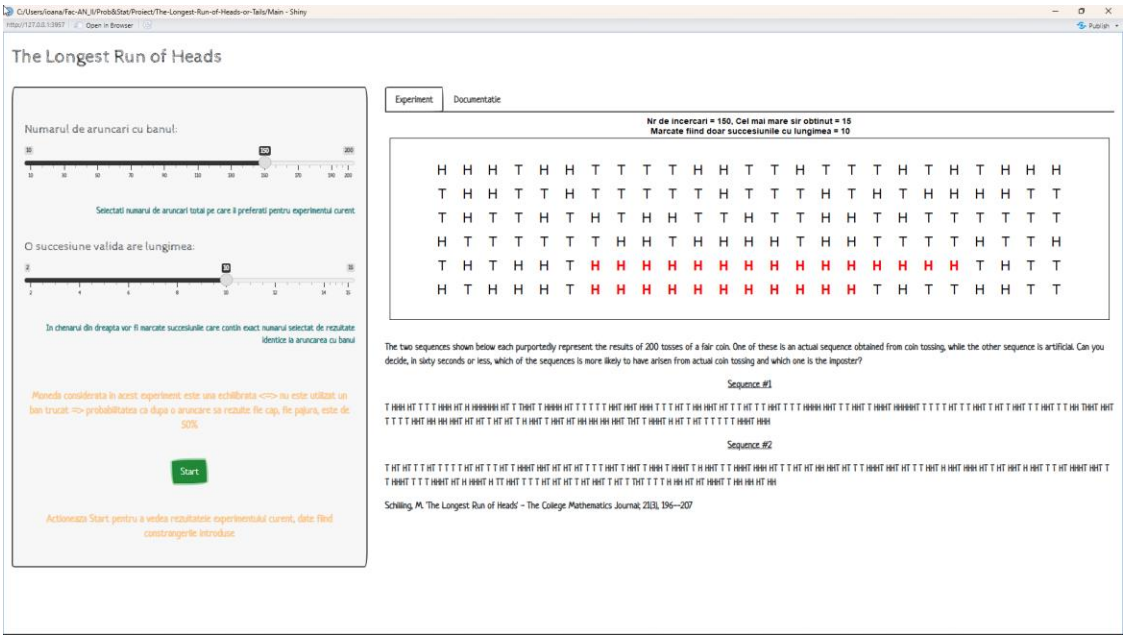
```
  if (i1 == i2) {
    mydata[val,3] <- all_Runs[curr]}
  else {
    curr <- curr + 1
    mydata[val,3] <- all_Runs[curr]}

  val <- val + 1
}

return(mydata)

}
```

```
plot.gen <- function(minlength, mydata){

  # we find the maximum length of a streak, observed during current
experiment
  maxlength <- max(mydata[,3])

  # we inform the user of the current experiment results
  exp.title <- paste("Nr de incercari = ", nrow(mydata), ", Cel mai mare sir
obtinut = ",
                     maxlength, "\n Marcate fiind doar succesiunile cu lungimea
= ", minlength, sep = "")


  tosses <- nrow(mydata)

  # we erase the white characters surrounding the plot, for better ui
experience
  par(mar=c(3,.5,3,.5)+0.1)

  # the plot consists of 25 columns and ceiling(tosses/25) rows
  plot(1,1,xlim=c(0,25+1),ylim=c(0,ceiling(nrow(mydata)/25)+1),col=0,
     yaxt="n",xaxt="n",xlab="",ylab="",main=exp.title)


  # we create a table containing the positions inside the matrix of each toss
  # x = the horizontal Cartesian coordinate of the coin flip [1..25]
  # y = the vertical Cartesian coordinate of the coin flip
[1..ceiling(tosses/25)]

  i <- seq(1, nrow(mydata)) # we need an index in order to arrange the
coordinates

  x <- i %% 25 # i = [0..24]

  x[which(x==0)]<-25 # x = [1..25], the horizontal structure of the matrix
```

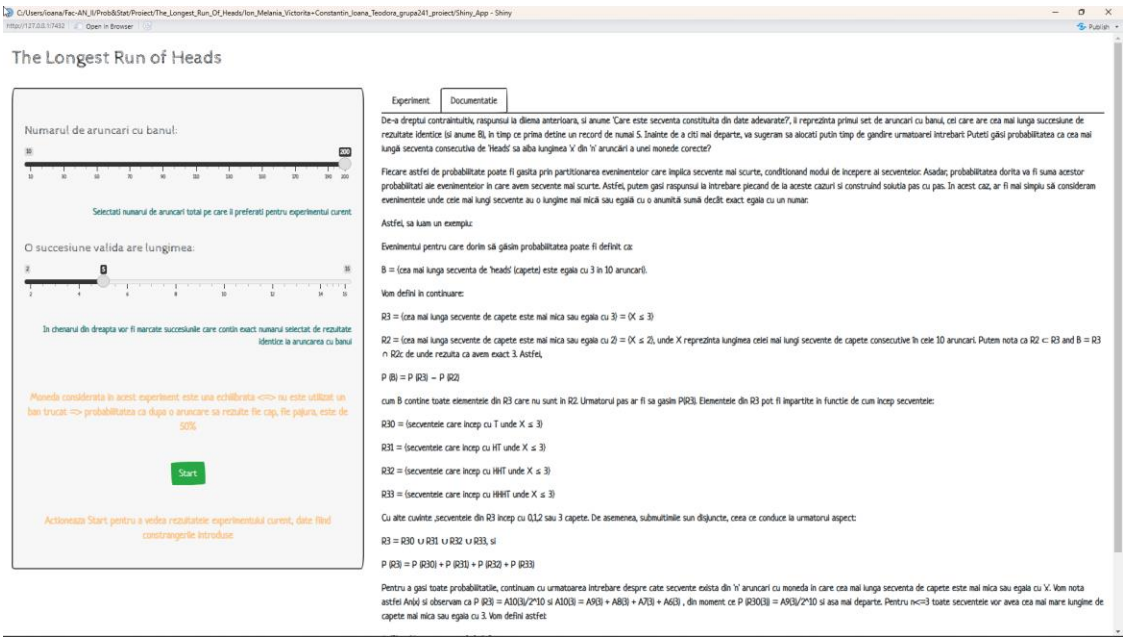
```
  # the vertical structure
  # it is reversed, as the streak of results is being written inside a plot
  # we use coordinates as if we were to write points inside a chart xOy
  y <- ceiling(nrow(mydata)/25) - ceiling (i/25) + 1

  # we bind the results together
  mydata <- cbind(mydata,x,y)


  # for each toss in particular
  for (index in 1:nrow(mydata)){
    # we verify if the result is "Heads", or "Tails"
    my.coin <-mydata[index,1]
    if(my.coin == 1){
      #Heads

      # we check if the current toss is part of a streak
      # if it is, we color it red
      if(mydata[index,3]>=minlength) text(mydata[index,4], mydata[index,5],
"H", col="red", font = 2, cex = 2)
      else text(mydata[index,4],mydata[index,5],"H", cex = 2)

    }

    if(my.coin == 0){
      # Tails
      text(mydata[index,4],mydata[index,5],"T", cex = 2)
    }

  }
}
```

Cum functioneaza aplicatia?

Pentru testul urmator, avand un nr de aruncari cu banul = 150 si o secventa maxima consemnata = 10, am obtinut urmatorul rezultat, in care cea mai mare secventa inregistrata este chiar 15:



Este, de asemenea, prezenta o zona de documentatie a proiectului, pentru a intelege ce se intampla in spatele acestui fenomen interesant.



1.”Schilling, M. ‘The Longest Run of Heads’,“The College Mathematics Journal”, 21(3), 196–207” 2.”The Longest Run of Heads” - Review by Amarioarei Alexandru (https://alexamarioarei.github.io/Research/docs/LongestHrunReview.pdf) 3.”Notes on the longest run of heads” - Bret Larget, September 18, 2009 (https://pages.stat.wisc.edu/~st309-1/heads-run.pdf)