

Discussion__week5

Salma Elshahawy

2/25/2020

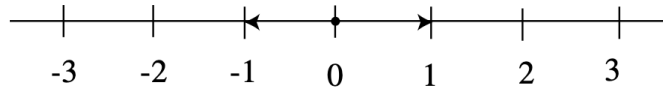
Mathematicians have been known to get some of the best ideas while sitting in a cafe, riding on a bus, or strolling in the park. In the early 1900s the famous mathematician George P'olya lived in a hotel near the woods in Zurich. He liked to walk in the woods and think about mathematics. P'olya describes the following incident:

At the hotel there lived also some students with whom I usually took my meals and had friendly relations. On a certain day one of them expected the visit of his fiancée, what (sic) I knew, but I did not foresee that he and his fiancée would also set out for a stroll in the woods, and then suddenly I met them there. And then I met them the same morning repeatedly, I don't remember how many times, but certainly much too often and I felt embarrassed: It looked as if I was snooping around which was, I assure you, not the case. This set him to thinking about whether random walkers were destined to meet. P'olya considered random walkers in one, two, and three dimensions. In one dimension, he envisioned the walker on a very long street. At each intersection the walker flips a fair coin to decide which direction to walk next (see Figure 1.6a). In two dimensions, the walker is walking on a grid of streets, and at each intersection he chooses one of the four possible directions with equal probability (see Figure). In three dimensions (we might better speak of a random climber), the walker moves on a three-dimensional grid, and at each intersection there are now six different directions that the walker may choose, each with equal probability

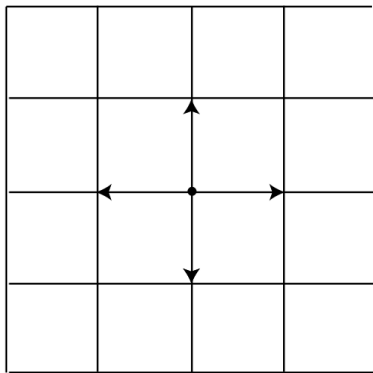
Write a program to simulate a random walk in one dimension starting at 0. Have your program print out the lengths of the times between returns to the starting point (returns to 0). See if you can guess from this simulation the answer to the following question: Will the walker always return to his starting point eventually or might he drift away forever?

```
# a random walk on the integers is a sequence  $X_0, X_1, \dots$  with  $X_0 = 0$ 
# and  $X_i = X_{i-1} + D_i$  where  $D_i$  is independent, equal to +1 with probability  $p$ , and otherwise -1
Random_walk <- function(x, size=10000){
  # This function is used to generate random walk
  # x : the point that you want to reach
  # size : how large the sample size you want to have at one time. This has default value 1000

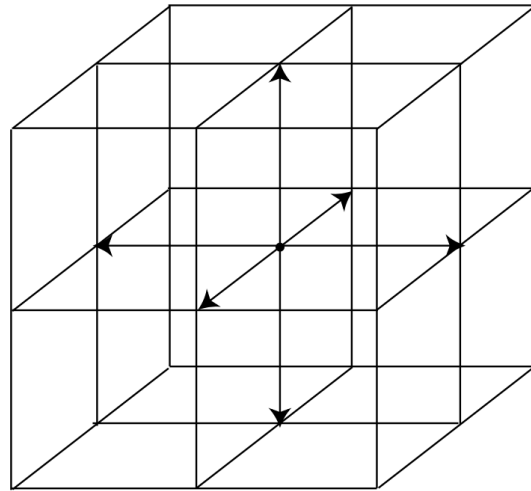
  # If the point is too large, try to choose larger sample size to shorten the time.
  set.seed(123) # To generate same result every time
  previous <- 0
  positions <- 0
  s <- 1
  final <- 0
  continue <- TRUE
  while(continue){
    # generate a vector for the move  $T(D_1, \dots, D_{size})$ 
    move <- sample(c(1,-1), size = size, TRUE)
    # Use cumsum to generate  $X$ 
    previous <- tail(final, 1) + cumsum(move)
    if(!is.na(which(previous == x)[1])){
      position <- which(previous == x)[1]
      cat("The total steps by far is: ", position + (s-1)*size, "\n")
      continue <- FALSE
    }else{
      positions[s] <- previous[size]
```



a. Random walk in one dimension.



b. Random walk in two dimensions.



c. Random walk in three dimensions.

Figure 1.6: Random walk.

Figure 1: random_walk

```
        final<-tail(positions,1)
        s <- s+1
    }
}
Random_walk(10, size = 10000)
```

The total steps by far is: 50

```
Random_walk(100, size = 10000)
```

The total steps by far is: 3130

```
Random_walk(1000, size = 10000)
```

The total steps by far is: 660482

From the simulation, I can see that the walkers would take infinite time and won't return to zero position. Where $X_n \rightarrow \text{infinity}$.