Levi Hamilton

CS-215-ON

Assignment 3.1 Recursive Dice

Let's examine the efficiency of the algorithm. Answer each of the following fully, explaining your logic and any supporting mathematics.

a. What is the probability that the die will land with a 2 showing?

Each roll is independent- therefore ⅙ or .167

b. In the best case, how many die rolls will occur? What is the algorithmic efficiency or Big O in that case?

Best case is that the algorithm will roll a 2 on the very first roll. Therefore, teh algorithmic efficiency with only one die roll is O(1).

c. What is the worst case? Are we concerned with the worst case?

Worst case is that the algorithm continues to roll the die indefinitely without rolling a 2. We are not really concerned with the worst case in this situation because technically there is no upper limit, but it is extremely unlikely the algorithm will never roll a 2.

d. What is the average case efficiency? Approximately how many die rolls should occur in the average case? Why do we care about the average case?

E(x)=(⅙)\*1+(⅚)\*(1+E(x))

=> 6E(x)=1+5\*(1+E(x))

=> 6E(x)=1+5+5E(x)

=> E(x)=6

Expected value is equal to 6, therefore on average it will take about 6 rolls to get a 2. We care about the average case because it shows the ‘normal’ behavior of the algorithm. Overall it helps us understand the performance if a large number of trials were necessary.

Part 3:

Update your program to run your roll class 1,000 times, recording the number of rolls it takes to get a 2 each time. Print the mean number of die rolls it takes to get a 2. Does it agree with your answer in Question 2.d? If not, why do you think it's different?

The updated program agrees with my answer from question 2.d. The result has yet to be exactly 6, byt hasnt reached a variation greater than 1 from just observation. The slight variation in the result is because the true random nature of the simulation.