Algorithms – Homework 2

1. To generate coin flips (Head or Tail 🡪 2 possibilities), we can split the six-sided die’s values to two 🡪 1,2,3 = head & 4,5,6 = tail
2. First flip for head ¾, second flip for head ¾ 🡪 ¾ \* ¾ = 9/16

First flip for tail ¼, second flip for tail ¼ 🡪 ¼\* ¼ =1/16

The probability that you will get nothing and have to try again after two flips is 9/19 + 1/16 = 10/16

1. If the coin is fair, then the first flip for head would be ½, second flip for head ½

First flip for tail ½, second flip for tail ½. So the probability that you will get not result after two flips is ¼ + ¼ = 1/2, or 50%

1. Roll the dice 6 times:

If on each role you get different number:

Output the first one

Otherwise:

Roll again

The efficiency of this algorithm depends on the how much biased it is. As the bias increases, the efficiency of the algorithm drops down.

1. For i from 0 to m-1:

j 🡪 random integer with 0 ≤ j ≤ i

Exchange a[j] and a[i]

The run time will be less than O(n) because we are picking m elements, and since it is stated that m<n, it means run time will be less than O(n) as well.

6. Randomize the deck of 52 cards, (like randomizing the list of 52 numbers) and then deal them to players, either by 1 or by 5 cards. Because it won’t change anything as long as the deck is randomized.

8. Places of A and B are exchanged, and the algorithm continues in the same way, as where the A is great than B.

9. Find the GCD of the two given numbers. Then divide the A\*B by the GCD

(A, B). The result will be the LCM (A, B).

7. Uploaded to github:

10. Step 1: Divide the exponent to powers of 2

Step 2: Calculate the mod of the powers of 2 until the given exponent is reached

Step 3: Combine the calculated mod values using modular multiplication rules

12. Uploaded to github:

From the results list, it does not seem that he number of step comparing to the average of the two numbers increases logarithmically.