ADw3

Name : **Peter Garamvoelgyi** Student ID: **2018280070**

**CLRS 5.2-3. Use indicator random variables to compute the expected value of the sum ofdice. (pg. 122)**

Letbe a random variable representing the sum ofdice.

Letbe a random variable representing the value of the’th dice.

As alldice are identical, we can replace these random variables with just one

Letbe an indicator random variable

The connection between these random variables is given by

Of course, exactly one ofwill take on the value 1.

We rely on the linearity of the expected value in our calculations:

And now we can simply calculate the final result:

**The expected value of the sum ofdice is: .**

**CLRS 5.2-5. Letbe an array ofdistinct numbers. Ifand, then the pairis called an inversion of A. Suppose that the elements ofform a uniform random permutation of. Use indicator random variables to compute the expected number of inversions. (pg. 122)**

Letbe a random variable corresponding to the number of inversions.

Let us introduce the indicator random variable

Given this definition, we can formulateas follows:

The expected value of a single indicator random variable is

Explanation: Let us partition the possibilities for the two numbers at positionsandbased on. Ifthen we haveoptions for. Ifthen we haveoptions for. The sum of these is given by. The rest of the numbers can be in any order:.

Using the linearity of the expected value, we can get the final solution:

**The expected value of the number of inversions is .**

**CLRS 5-2. Searching an unsorted array (pg. 143)**

**a. Write pseudocode for a procedure Random-search to implement the strategy above. Be sure that your algorithm terminates when all indices intohave been picked.**

**Input**: A sequence ofnumbers and a value

**Output**: An indexs.t.or the special value nil ifdoes not appear in.

Random-search:

1

2

3 **loop forever**

4 Random

5 **if** **then**

6 **return**

7 **if** **then**

8

9

10 **if** **then**

11 **return** nil

12 **end loop**

**b. Suppose that there is exactly one indexsuch that. What is the expected number of indices intothat we must pick before we findand Random-search terminates?**

Let us introduce the indicator random variable

… characterized by a geometric distribution.

Letbe a random variable corresponding to the number of indices we must pick.

**c. Generalizing your solution to part (b), suppose that there areindicessuch that. What is the expected number of indices intothat we must pick before we findand Random-search terminates? Your answer should be a function ofand.**

Let us introduce the indicator random variable

**d. Suppose that there are no indicessuch that. What is the expected number of indices intothat we must pick before we have checked all elements ofand Random-search terminates?**

The question is: How many indices do we have to draw until we checked all of them?

The probability of checking an unchecked entry once we have checkedis

whereis a random variable with a geometric distribution.

The number of draws required until we have checked all entries is

(Following the derivation in CLRS 5.4.2)

**e. Suppose that there is exactly one indexsuch that.What is the average-case running time of Deterministic-search? What is the worst-case running time of Deterministic-search?**

Deterministic-search:

1 **for**  **to**

2 **if** **then**

3 **return**

4 **return** nil

Let us introduce the indicator random variable

Letbe a random variable corresponding to the number of indices we must check.

We can estimate the average-case running time using the expected value:

The worst-case running time is when the target element is the last in the array. In this case we needcomparisons.

**CLRS 8.4-4. We are givenpoints in the unit circle,, such thatfor. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design an algorithm with an average-case running time ofto sort thepoints by their distancesfrom the origin. (*Hint*: Design the bucket sizes in Bucket-sort to reflect the uniform distribution of the points in the unit circle.) (pg. 204)**

Let us partition the circle intoconcentric rings of equal area. The area of the circle is . The area of the’th ring is given by the formula

… whereis the outer radius andis the inner one.



Such a partitioning has multiple desirable properties:

1. As the points are uniformly distributed within the circle, the probability that a given point is in ringis simply.
2. The distance of points within ringfrom the origin is larger than the distance of points within ring.

Let us use these rings as our buckets for Bucket-sort. Our algorithm becomes:

1. Iterate through all points and put each into the correct bucket
2. Sort the points in each bucket based on their distances from the origin.
3. Reconstruct the sorted result by iterating through rings

**If we choosethen the average complexity of this algorithm is.**