D2 : Experiments and stacks

A cereal maker puts a card with a sports figure in each of the cereal boxes. There are 12 figures, and each box contains a card of a uniformly random figure. Little Joe wants to collect all the 12 cards. How many boxes of cereal must he eat? This is known as the *coupon collector problem.* Your task is to experimentally estimate this number (for N cards).

a) Write a method int couponCollectorTest(int N) that performs one such experiment. There are *N* „cards" to be collected, which we can number as 0 to *N*- You want to generate random integers in that range until all *N* numbers have appeared. The method outputs how many numbers were generated in the process (in total).

Note: You need to stop generating as soon as all numbers are generated, but without spending much time. The method should take time proportional to the numbers generated.

Test your method with N=1000 and N=10000, three times each.

b) You notice from your tests in part a that the time spent can vary a lot. The task now is to examine this statistically by repeating the experiment. Given numbers N and T, you want to perform the experiment above with N cards, and repeat it T times. You then want to examine the statistics of the counts generated.

Implement a class with the following API:

```
public class CouponCollectorStats  {
  public CouponCollectorStats(int N, int T)
  public double mean()
  public double stddev()
}
```

The constructor performs T coupon collector experiments, each involving N cards. The methods mean() and stddev() output the mean and standard deviations of the counts. As always, you may use any part of the algs4 codebase as part of your solution. It is also practical to use your solution for a) here.

c) Add a unit-test main() function that tests the class with different values of N and T. In particular, it should perform the combined experiment with N=1000 and T ranging as 10, 100, 1000, 10000, indicating both the mean and the standard deviations of the counts.

Example with N=100, T=3: Running couponCollectorTest(100) three times gave as output 680, 510, 460. (That is, in the first experiment, we had to generate 680 random numbers until all had appeared.) The mean is then (680+510+460)/3 = 550.

The class should be documented with Javadoc. In particular, there should be general documentation for the class, and documentation of each method/constructor detailing parameters, functionality and return value.

d) Find (e.g., by internet search) the theoretic estimate for coupon collection. Compare your bounds with this theory, for different N and T. What effect does varying T have?

Turn in:

1. a) The code (in a single class)
2. b) Javadoc of the class, and
3. c) The output of the experiments (as described above)

Note: These should be separate files, not as a ZIP file.

Class Problems

These are problems to be discussed in class. They are not for credit, but you are encouraged to attempt to solve them before coming to class.

I. *Dæmi 1.3.3 á bls 161*: Segjum að notandaforrit framkvæmi blöndu af push og pop aðgerðum á stafla. Push aðgerðirnar ýta heiltölunum 0 til 9 á staflann í þeirri röð; pop aðgerðirnar prenta út gildin. Hverjar af eftirfarandi úttaksrunum geta ekki birst?

Problem 1.3.3 / Suppose that a client performs an intermixed sequence of (stack) push and pop operations. The push operations put the integers 0 through 9 in order onto the stack; the pop print out the return values. Which of the following sequence(s) could not occur?

1. a) 4 3 2 1 0 9 8 7 6 5
2. b) 4 6 8 7 5 3 2 9 0 1
3. c) 2 5 6 7 4 8 9 3 1 0
4. d) 4 3 2 1 0 5 6 7 8 9
5. e) 1 2 3 4 5 6 9 8 7 0
6. f) 0 4 6 5 3 8 1 7 2 9
7. g) 1 4 7 9 8 6 5 3 0 2
8. h) 1 2 3 4 5 6 7 8 9 0

II. (1.3.9) Write a program that takes from standard input an expression without left parenthesis and prints the equivalent infix expression with the parentheses inserted. For example, given the input

1 + 2) * 3 – 4)

your program should print

((1+2)*(3-4))