BLG 335E ANALYSIS OF ALGORITHMS

HW#1

Due Date: 29 October 2019, 23:00

Homework Policy

- Use comments whenever necessary to explain your code. Also, **write your name on top of each file you are sending in comment!**
- Do not forget to handle exceptions and print error messages!
- Your code must be written in C++, and should be able to be compiled and run using g++ compiler. ITU servers provide g++ compiler, you can test your program by connecting to the servers using ssh (if you don't know how to do it, click here).
- IMPORTANT: This is an individual assignment! You are expected to act according to Student Code of Conduct, which forbids all ways of cheating and plagiarism. It is okay to discuss the homework with others, but it is strictly forbidden to use all or parts of code from other students' codes or online sources and let others do all or part of your homework. **Your codes will be checked using plagiarism detection software. In case of cheating occurs, you will be subject to disciplinary actions.**
- If you have any questions, write it to **HW1 message board** on Ninova.

Problem Description

Let **A[1..n]** be an array of **n** numbers. If i<j and A[i]=A[j], then the pair **(i,j)** is called a ***duplication*** of A.

Suppose that the elements of A are filled in by a uniform random number generator that produces numbers in the range **[1..n]** (i.e., **1 ≤ A[i] ≤ n** for **i =1 to n**).

In this homework, you are asked to compute expected number of duplications for an array of size **n** with paper and pencil and then verify your calculation by a C++ program implementation, as follows:

(a) Use indicator random variables to compute the expected number of duplications for an **array of n random numbers**, as described above.

(b) Write a C++ program that

   a. takes number of array elements (**n**) from the command line.
   b. fills in the array **with random numbers** from the range **[1..n].**
   c. outputs array elements (for clarity, output up to 10 array elements per line).
   d. outputs **expected number of duplications**, based on your analysis in Part (a).
   e. finds all the duplications of the array and prints them out.
   f. outputs **encountered number of duplications**, based on the program run.

Two example program runs are given below:

```
E:\>aoa_hw1.exe 30
ARRAY:
25   13   28   7    24   18   22   20   16   26
21   29   15   3    7    25   16   4    17   3
24   21   28   21   7    30   7    14   10   12

EXPECTED NUMBER OF DUPLICATIONS = ... (WRITE HERE THE EXPECTED NUMBER OF DUPLICATIONS, BASED ON YOUR ANALYSIS)

(1,16) (3,23) (4,15) (4,25) (4,27) (5,21) (9,17) (11,22) (11,24) (14,20)
(15,25) (15,27) (22,24) (25,27)

ENCOUNTERED NUMBER OF DUPLICATIONS = 14
```

Figure 1: An Example Program run with array size n = 30

```
E:\>aoa_hw1.exe 100
ARRAY:
48   55   23   37   85   16   42   88   65   75
65   2    17   69   3    32   48   31   69   18
13   26   35   63   70   33   78   8    18   78
79   54   22   63   28   44   40   26   81   27
52   79   9    68   27   49   50   81   80   28
87   16   45   66   9    68   71   75   92   38
50   53   50   31   7    7    64   6    54   81
8    36   50   85   99   31   95   49   75   20
11   9    20   60   56   70   3    97   29   64
51   80   23   79   35   34   38   4    7    51

EXPECTED NUMBER OF DUPLICATIONS = ... (WRITE HERE THE EXPECTED NUMBER OF DUPLICATIONS, BASED ON YOUR ANALYSIS)

(1,17) (3,93) (5,74) (6,52) (9,11) (10,58) (10,79) (14,19) (15,87) (18,64)
(18,76) (20,29) (22,38) (23,95) (24,34) (25,86) (27,30) (28,71) (31,42) (31,94)
(32,69) (35,50) (39,48) (39,70) (40,45) (42,94) (43,55) (43,82) (44,56) (46,78)
(47,61) (47,63) (47,73) (48,70) (49,92) (55,82) (58,79) (60,97) (61,63) (61,73)
(63,73) (64,76) (65,66) (65,99) (66,99) (67,90) (80,83) (91,100)

ENCOUNTERED NUMBER OF DUPLICATIONS = 48
```

Figure 2: An Example Program run with array size n = 100

Report

Your report must consist of two parts:

(1) In the first part, you must present your analysis for **Part (a)** above, i.e., use of indicator random variables to compute the expected number of duplications.

(2) In the second part, analyze your program that is implemented in **Part (b)** and calculate the running time of the algorithm.

Submission

Submit your homework files through Ninova. Please zip and upload all your files. You are going to submit the following files:

(a) All your .h (if any) and .cpp files

(b) A .pdf file as your report. Your report should be clear and detailed. If your solutions are unclear, it may result in a grade loss for you.

(c) A .txt file explaining how to compile and run your code (in a 1 or 2 line)

Use your student ID as file names. That is, **your_student_id.zip** for the .zip file, **your_student_id.cpp** for the main C++ source file and **your_student_id.pdf** for the report.