Computer Science CSC263H
St. George Campus

February 25, 2016
University of Toronto

Homework Assignment #4

Due: March 10, 2016, by 5:30 pm

- You must submit your assignment as a PDF file of a typed (**not** handwritten) document through the MarkUs system by logging in with your CDF account at `markus.cdf.toronto.edu/csc263-2016-01`. To work with a partner, you and your partner must form a group on MarkUs.
- The PDF file that you submit must be clearly legible. To this end, we encourage you to learn and use the LaTex typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You can use other typesetting systems if you prefer, but handwritten documents are not accepted.
- By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.[a]
- For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the assigned sections of the official course textbook, by referring to it.
- Unless we explicitly state otherwise, you should justify your answers. Your paper will be marked based on the correctness and completeness of your answers, and the clarity, precision, and conciseness of your presentation.

[a] "In each homework assignment you may collaborate with at most one other student who is currently taking one of the sections of CSC263H taught this term. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. **Collaboration involving more than two students is not allowed. For help with your homework you may consult only the course instructors, teaching assistants, your homework partner (if you have one), your textbook and your class notes. You may not consult any other source.**"

Question 1. (20 marks)

Let $p_2, \ldots, p_n$ be real numbers in $[0, 1]$, to be specified later. Consider the following randomized algorithm:

```
x = 1
for i = 2 to n
        With probability p_i, x = i; otherwise x is unchanged.
return x
```

a. Consider the value of x returned by the above procedure. Compute $\Pr[x = i]$ for each $1 \le i \le n$, under the assumption that $p_2 = p_3 = \ldots = p_n = 1/2$. Give a precise mathematical derivation of these probabilities.

b. Give values for $p_2, \ldots, p_n$ so that for any $1 \le i \le n$, $\Pr[x = i] = 1/n$, i.e. x is a uniform sample from $1, \ldots, n$. Prove that, with the values of $p_2, \ldots, p_n$ that you give, the probability of $x = i$ is $1/n$, as required.

Question 2. (20 marks) Assume you have a biased coin, which, when flipped, falls on Heads with probability p, and on Tails with probability $1 - p$. However, you do not know p. How can you use the coin to simulate an unbiased coin? Formally, you have access to a procedure FlipBiasedCoin(), which returns either 1 or 0 at random. FlipBiasedCoin() returns 1 with probability p, and 0 with probability $1-p$, but you do not know p. Design an algorithm that, without making any other random choices except calling FlipBiasedCoin(), returns 1 with probability 1/2 and 0 with probability 1/2. Your algorithm should not use p. You can assume that each time you call FlipBiasedCoin(), the value it returns is independent of all other calls to it.

a. Describe the algorithm in clear and concise English, and prove that it outputs 0 with probability 1/2 and 1 with probability 1/2.

b. Analyze the expected running time of your algorithm. Note that while the algorithm itself does not use p, the expected running time should be expressed in terms of p.

Question 3. (20 marks)

Let $V = \{1, \ldots, n\}$. We are given as input a sequence of edges $e_1, \ldots, e_m$, where for each $1 \le i \le m$, $e_i = (j, k)$ for some $j, k \in V$, $j \ne k$. For each $0 \le i \le m$, define the undirected graph $G_i = (V, E_i)$ where $E_0 = \emptyset$, and for $i \ge 1$, $E_i = \{e_1, \ldots, e_i\}$. Design an algorithm that outputs an array $C[0\,..\,m]$, with $C[i]$ equal to the number of nodes of the largest connected component of G_i (note that $C[0] = 1$). Your algorithm should run in time $O(n + m \log^*(n))$.

Describe the algorithm in clear and concise English. Explain why it is correct and why it runs in the required time complexity.