CSC473 W18: Homework Assignment #2

Due: March 14, by midnight

Guidelines: (read fully!!)

- Your assignment solution must be submitted as a *typed* PDF document. Scanned handwritten solutions, solutions in any other format, or unreadable solutions will **not** be accepted or marked. You are encouraged to learn the LaTeX typesetting system and use it to type your solution.
- To submit this assignment, use the MarkUs system, at URL `https://markus.teach.cs.toronto.edu/csc473-2018-01`
- This is a *group assignment*. This means that you can work on this assignment with *at most one other* student. You are *strongly encouraged* to work with a partner. Both partners in the group should work on and arrive at the solution together. Both partners receive the same mark on this assignment.
- Work on all problems together. For each problem, one of you should write the solution, and one should proof-read and revise it. The first page of your submission must list the *name*, *student ID*, and *UTOR email address* of both group members. It should also list, for each problem, which group member wrote the problem, and which group member proof-read and revised it.
- You **may not** consult any other resources except: your partner; your class notes; your textbook and assigned readings. *Consulting any other resource, or collaborating with students other than your group partner, is a violation of the academic integrity policy!*
- You may use any data structure, algorithm, or theorem previously studied in class, or in one of the prerequisites of this course, by just referring to it, and without describing it. This includes any data structure, algorithm, or theorem we covered in lecture, in a tutorial, or in any of the assigned readings. Be sure to give a *precise reference* for the data structure/algorithm/result you are using.
- Unless stated otherwise, you should justify all your answers using rigorous arguments. Your solution will be marked based both on its completeness and correctness, and also on the clarity and precision of your explanation.

Question 1. (10 marks) In this question you will design a Markov chain to approximately sample a permutation of the integers $[n] = \{1, \dots, n\}$. Given a permutation $\sigma = \sigma_1, \dots, \sigma_n$ of $[n]$, let $\mathrm{inv}(\sigma)$ be the number of distinct pairs $i < j$ such that $\sigma_i > \sigma_j$. I.e. $\mathrm{inv}(\sigma)$ is the number of pairs of integers that are inverted in σ. Consider a graph $G = (V, E)$ where V is the set of *all* permutations of $[n]$, and there is an edge in E from σ to σ' if we can get σ' by swapping two numbers in σ. Additionally, every vertex has an edge to itself.

a. Prove that the graph G is strongly connected.

b. Using the Metropolis-Hastings Algorithm, give the transition probabilities of a Markov chain on G so that in the stationary distribution p of the chain the probability p_σ of σ is proportional to $2^{-\mathrm{inv}(\sigma)}$.

c. Implement a single step of the Metropolis-Hastings Algorithm from the previous subquestion. Your algorithm should take a permutation σ, stored as an array of n cells, and sample a random permutation σ' according to the transition probabilities prescribed in the previous subquestion. Your algorithm should work in time and space polynomial in n. Note that this does *not* allow you to store the graph G explicitly.

Question 2. (10 marks) Consider the following streaming algorithm, which takes a stream $\sigma = \sigma_1, \dots, \sigma_m$, with $\sigma_t \in [n]$ for each $1 \le t \le m$:

```
TOW($\sigma$)
1  Choose random hash functions $h_1, \dots, h_k$ from $[n]$ to $\{-1, 1\}$
2  Initialize $c_1 = \dots = c_k = 0$
3  for $t = 1$ to $m$
4      for $j = 1$ to $k$
5          $c_j = c_j + h_j(\sigma_t)$
6  Output $c = \frac{1}{k}(c_1^2 + \dots + c_k^2)$.
```

Assume the random hash functions $h_1, \dots h_k$ are independent. Moreover, for each $1 \le j \le k$ assume that the values $h_j(1), h_j(2), \dots, h_j(n)$ are independent, and each of them is equally likely to be equal to -1 or $+1$.

a. What is $\mathbb{E}[c]$? Give your answer as a simple expression in terms of the frequencies $f_i = |\{t : \sigma_t = i\}|$. Justify your answer.

b. Assume the inequality $\mathbb{E}[c_j^4] \le 3\mathbb{E}[c_j^2]^2$ holds for each $1 \le j \le k$. Give a value of k such that

$$(1 - \varepsilon)\mathbb{E}[c] \le c \le (1 + \varepsilon)\mathbb{E}[c]$$

holds with probability at least $\frac{1}{2}$. Give the smallest such value of k that you can. Justify your answer.

Bonus: Prove that $\mathbb{E}[c_j^4] \le 3\mathbb{E}[c_j^2]^2$ holds for all j.

Question 3. (10 marks) You are given as input a linear program

$$\max c^\intercal x \tag{1}$$
$$\text{s.t.} \tag{2}$$
$$Ax \le b \tag{3}$$
$$x \ge 0 \tag{4}$$

Give an algorithm that takes A, b, c as input, and constructs in polynomial time a matrix D and vectors e, f such that the linear program

$$\max f^\intercal y$$
$$\text{s.t.}$$
$$Dy \le e$$

has the same optimal value as (1)–(4). Moreover, each row of D should have at most three nonzero entries. Finally, restricting the variables vector y to a subset of its coordinates should give an optimal feasible solution to (1)–(4).

Precisely describe the new linear program, and how to generate it from the input linear program, and argue why it satisfies the required properties.