ECE 606, Fall 2021, Assignment 8
Due: Tuesday, November 9, 11:59pm

Submission: submit three things: (i) your solutions for the written problems to crowdmark, (ii) the page with your name and student ID for the **[python3]** problem to crowdmark, and, (iii) your **[python3]** file to the appropriate Dropbox on Learn.

1. Call an undirected graph G *randomly sparse* if given a set of vertices $\{u, v\}$ of cardinality exactly 2, the probability that the edge $\langle u, v\rangle$ exists in G is $\frac{1}{10}$.

   How many edges do we expect a randomly sparse G has, assuming G has n vertices?

2. **[python3]** Suppose you are given as input (i) an undirected graph $G = \langle V, E\rangle$ and (ii) a positive integer k. Now suppose we generate a *random assignment function* $f : V \to \{1, 2, \dots, k\}$ as follows: for every $v \in V$ and $i \in \{1, \dots, k\}$, $\Pr\{f(v) = i\} = 1/k$. That is, f associates an integer $\in \{1, 2, \dots, k\}$ uniformly at random with each vertex in G.

   Call an edge $\langle u, v\rangle \in E$ *satisfied* under f if $f(u) \neq f(v)$. (Underlying this is the graph colouring problem, and ideally, we would like all edges to be satisfied, i.e., two vertices that have an edge between them to receive different colours. In this problem, we want to analyse the simple algorithm of randomly assigning colours to each vertex.)

   We want to compute, empirically, the expected number of edges in G that are satisfied. Specifically, write a python subroutine `avgsatisfied` in `a8p2.py` whose skeleton is provided on Learn that takes input undirected $G = \langle V, E\rangle$ encoded as an adjacency list and a positive integer k and works as follows.

   Initialize a running average a
   repeat
   $f \leftarrow$ a new function with domain V and codomain $\{1, \dots, k\}$ as specified above
   $c \leftarrow$ number of edges in G satisfied under f
   Incorporate c into the running average a
   until *5000 times*
   return a

   Not required explicitly for the assignment: you may want to work out for yourself analytically what the expected number of satisfied edges is, and check that it agrees with what your subroutine returns. That is exactly how we will test your submission. See the tester file provided to you on Learn for some examples.

3. Given non-empty finite sets A, B, we define a *random function* with domain A and codomain B as follows: from the set of all functions each with domain A and codomain B, we pick a function uniformly at random.

   Prove that a random function $f : U \to \{0, \dots, m-1\}$ meets the simple uniform hashing assumption. (See pdf page 11 of Lecture 8 of the textbook for what that assumption is.)

   (Hint: you may first want to determine how many functions exist with domain A and codomain B as a function of $|A|$ and $|B|$. For this, think of each function as an array of $|A|$ entries, each of which is the member from B to which a member from A is mapped.)

4. Recall from Definition 2 of pdf page 7 of Chapter 3 of your textbook that a *vertex cover* of an undirected graph $G = \langle V, E \rangle$ is any $C \subseteq V$ such that $\langle u, v \rangle \in E \implies ((u \in C) \vee (v \in C))$. An *independent set* of an undirected graph $G = \langle V, E \rangle$ is any set $I \subseteq V$ such that $((u \in I) \wedge (v \in I)) \implies \langle u, v \rangle \notin E$.

   (a) Prove that C is a vertex cover of G if and only if $V \setminus C$ is an independent set of G.

   (b) Alice proposes the following algorithm ApproxIndSetSize for computing an approximation for the size of a maximum-sized independent set of an undirected graph G that invokes ApproxVertexCover from pdf page 7 of Lecture 3 of your textbook as a subroutine.

   ApproxIndSetSize($G = \langle V, E \rangle$)
   $C \leftarrow$ ApproxVertexCover(G)
   if $C = V$ **then return** $|V|/2$
   else return $|V \setminus C|$

   Recall from Lecture 3 of your textbook that ApproxVertexCover is guaranteed to return a vertex cover whose size is no more than double the size of a minimum-sized vertex cover of G.

   Alice claims that ApproxIndSetSize is guaranteed to output a value which is no less than half that of a maximum-sized independent set of G. Is Alice's claim true? Begin your response with 'yes' or 'no,' then prove/disprove accordingly.