CSE 21 Spring 2022
Homework 6
Due date: Monday, May 23, 2022 at 11:59pm

INSTRUCTIONS

This homework should be done in **groups** of one to four students, without assistance from anyone besides the instructional staff and your group members. Homework must be submitted through Gradescope by a **single representative** of your group and received by **11:59pm** on the due date. There are no exceptions to this rule.

You will be able to look at your scanned work before submitting it. You must **type** your solutions. (hand-drawn diagrams are okay.) Your group representative can resubmit your assignment as many times as you like before the deadline. Only the most recent submission will be graded.

Students should consult their textbook, class notes, lecture slides, podcasts, group members, instructors, TAs, and tutors when they need help with homework. You may ask questions about the homework in office hours, but questions on Piazza should be private, visible only to instructors.

This assignment will be graded for not only the *correctness* of your answers, but on your ability to present your ideas clearly and logically. You should explain or justify, present clearly how you arrived at your conclusions and justify the correctness of your answers with mathematically sound reasoning (unless explicitly told not to). Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to **convince the reader** that your results and methods are sound.

KEY CONCEPTS recursive counting, graph modeling, graph algorithms.

---

1. (12 points) An RNA string is a string over the alphabet $\{A, C, G, U\}$.

   (a) Let $X(n)$ be the number of RNA strings of length n that avoid the substring AU.

      i. Show that$X(0) = 1, X(1) = 4$ and $X(n) = 4X(n-1) - X(n-2)$ for all $n \geq 2$.

      ii. Determine the asymptotic class that $X(n)$ belongs to. (Show your work.)

   (b) Let $Y(n)$ be the number of RNA strings of length n such that A never appears anywhere after U.

      i. Determine a recurrence relation for $Y(n)$. (Don't forget the base case(s) and explain your answer.)

      ii. Determine the asymptotic class that $Y(n)$ belongs to. (Show your work.)

2. (12 points) Let $C(n)$ be the nth complete DAG defined recursively as follows.

   $C(0)$ is a single vertex x_0.

   $C(n)$ is the result of taking $C(n-1)$, adding in a new vertex x_n and creating a directed edge from x_n to all vertices of $C(n-1)$.

   (a) Let $E(n)$ be the total number of edges of $C(n)$. Determine a recurrence relation for $E(n)$ and solve for its closed form. (Don't forget the base case(s).)

   (b) Let $P(n)$ be the total number of paths from x_n to x_0 in $C(n)$. Determine a recurrence relation for $P(n)$ and solve for its closed form. (Don't forget the base case(s).)

3. (16 point) UNO is a game played with a deck of cards consisting of cards that have a number from 0 to 9 and a color from the set {red, blue, yellow, green}. (In this version there are no special cards like reverse and draw four.)

   A valid pile is an ordering of the cards such that each pair of consecutive cards either share the same number or share the same color (or both.)

   For Example:

   $(7G, 6G, 0G, 0B, 4B, 4R, 0R)$ is a valid pile of uno cards.

   You are given a collection of n uno cards (with possible repeated cards.) You want to answer the question: "is it possible to arrange all n uno cards into a single valid pile of n cards.

   For example: the collection: $\{0R, 0B, 0G, 4R, 4B, 6G, 7G\}$ can be arranged so that all n cards are in a single valid pile of n cards (See the valid pile above.)

   On the other hand, the collection: $\{0R, 0B, 0G, 4R, 4B, 6G, 7Y\}$ cannot be put into a single valid pile.

   (a) Model this problem as a graph problem in such a way that, given the collection of n uno cards, build a graph G such that G has a Hamiltonian path if and only if the collection of n cards can be put into a single valid pile.

      (Your answer should be a description of a graph for a general collection of cards. You must have a clear description of the vertices and a clear rule about when two vertices are connected by an edge and whether or not the edge is directed or undirected.)

   (b) If there are n cards, how many vertices and edges does your graph have (worst-case using Big-O notation)

   (c) Draw the graph for the input:

      $\{4Y, 9R, 5Y, 4R, 6G, 8G, 6R, 8B, 5G, 0Y, 0B\}$

      (Try your best to draw your graph as a *planar* graph meaning that none of the edges cross.)

   (d) Identify a Hamiltonian path in the graph you built for part c and use that to construct a valid pile.

4. Recall that adjacency matrices of *simple* directed graphs are matrices consisting of 0's and 1's such that there are no 1's down the diagonal (no self loops, no parallel edges.)

   (Review of matrix multiplication can be found on page 179 of Rosen.)

   A directed 3-cycle in a simple directed graph is a set of three distinct vertices x, y, z such that $(x, y) \in E, (y, z) \in E$ and $(z, x) \in E$.

   (a) (20 points) Consider the following algorithm that takes as input an adjacency matrix of a simple directed graph G and returns True if there exists a directed 3-cycle and returns False if there is not a directed 3-cycle.

   3-cycle1(G) (G, a simple directed graph with n vertices in adjacency matrix form.)

   ```
   for i = 1, ..., n:
       for j = 1, ..., n:
           if G[i, j] == 1 :
               for k = 1, ..., n:
                   if G[j, k] == 1 and G[k, i] == 1:
                       return True
   return False
   ```

   (4 points) Show that the runtime for this algorithm is $O(n^2 + n|E|)$ where $|E|$ is the number of edges in the graph.

   (b) (4 points) In order for *3-cycle1* to return True, what needs to happen and why does this correspond to the graph having a directed 3-cycle?

   (c) (4 points) Show that if G is the adjacency matrix of a simple directed graph and $H = G^2$, then $H[i, j]$ is the number of paths of length 2 from i to j.

   (d) Consider the following algorithm that takes as input an adjacency matrix of a simple directed graph G and returns True if there exists a directed 3-cycle and returns False if there is not a directed 3-cycle.

   3-cycle2(G) (G, a simple directed graph with n vertices in adjacency matrix form.)

   ```
   Compute H = G^2                                   # matrix multiplication
   for i = 1, ..., n:
       for j = 1, ..., n:
           if G[i, j] == 1 AND H[j, i] > 0:
               return True
   return False
   ```

   (4 points) Assuming that computing the square of a $n \times n$ matrix takes $O(n^{2.81})$ time, calculate the runtime of this algorithm.

   (e) (4 points) Is *3-cycle1* or *3-cycle2* more efficient or does it depend on the input? (Justify your answer.) (Hint: think about dense and sparse graphs.)