Name _____

CSCI 332, Fall 2025
Exam 3

Note that this exam has two sections. They are:

1. 2D Dynamic Programming (50 points)
2. Complexity Classes (P, NP, and NP-Completeness) (50 points)

You may use a double sided 3x5 handwritten notecard of notes during the test but no other resources. If you need more space than what is given, develop your solution on scratch paper before copying your final answer to the exam paper.

Good luck!

Section 1 (2D Dynamic Programming)

1. In preparation for the University of Montana CS department's annual winter bowling party (6-7:30pm on Saturday, December 13th, 2025 at Westside Lanes), you have been practicing with the following, simpler version of bowling. There are n pins in a single row, numbered from 1 to n. Each pin i has a point value $P[i]$ associated with it. You can knock down either one single pin or two adjacent pins each time you roll the ball. Because you have been practicing so much, you always hit the pins you aim for. If you knock down pin i, you earn $P[i]$ points. If you knock down pins i and $i+1$, you earn the product $P[i] \cdot P[i+1]$ points.

   You want to maximize your score by choosing which pins to knock down.

   For example, if you have 4 pins with values $P = [2, 3, 5, 1]$, one optimal strategy is to knock down pin 1 alone, pins 2 and 3 together, and pin 4 alone, earning $2 + (3 \cdot 5) + 1 = 18$ points.

   (a) (10 points) Write the *English definition* of the subproblems you will solve in order to write a dynamic programming algorithm for this problem.

   (b) (10 points) Write the *recursive definition* of the subproblems you will solve in order to write a dynamic programming algorithm for this problem. Be sure to include all necessary base cases!

   (c) (5 points) How will you memoize your subproblems?

(d) (5 points) Write a pseudocode for the iterative, dynamic programming algorithm to compute the maximum score you can achieve.

(e) (bonus! skip if not enough time) Give an example input to the simple bowling problem for which a *greedy strategy* of pairing adjacent pins with the highest product fails to find the optimal solution.

2. (20 points) Recall the *edit distance* problem from lecture. In this problem, you are given two strings $A[1..m]$ and $B[1..n]$, and you want to transform A into B using the following operations:

   - Insert a character into A
   - Delete a character from A
   - Substitute a character in A for another character

   The edit distance between strings A and B is the minimum number of such operations needed to transform A into B.

   We can solve this problem using dynamic programming. Let $Edit(i, j)$ be the edit distance between the first i characters of A and the first j characters of B.

   Fill in the base case(s) for the following recursive definition of $Edit(i, j)$. (Let $A[i] \neq B[j]$ be 1 if the characters are different and 0 if they are the same.)

$$Edit(i,j) = \begin{cases} \\ \\ \\ \\ \\ \min \left\{ \begin{array}{c} Edit(i-1,j-1) + A[i] \neq B[j], \\ Edit(i-1,j) + 1, \\ Edit(i,j-1) + 1 \end{array} \right\} & \text{if } i,j > 0 \end{cases}$$

Section 2 (Complexity Classes)

3. (16 points) Assuming $P \neq NP$, draw a Venn diagram showing the relationships between the following sets of decision problems:

   - P
   - NP
   - NP-Complete
   - NP-Hard

   The relative sizes of the sets do not matter, but the intersections and containments must be correct.

4. (14 points) For each of the following decision problems, circle the sets that the problem belongs to.

   (a) *Is-Sorted*: Given an array of n integers, determine whether the array is sorted in non-decreasing order.
   P NP NP-Complete NP-Hard

   (b) *Min-Vertex-Cover*: Given an undirected graph $G = (V, E)$ and an integer k, determine whether k is the smallest possible size of a subset X of V such that every edge in E is incident to at least one vertex in X.
   P NP NP-Complete NP-Hard

5. Still excited for the bowling party, you are considering the following problem.

   You will bowl exactly n frames and you are given two length n arrays of integers A and B. In each frame i, you can score either $A[i]$ points or $B[i]$ points. (With all of your practice, you can always achieve either score you choose.) You want to know whether there is a way to choose your scores in each frame so that your total score is exactly T points.

   For example, if $n = 3$, $A = [2, 3, 5]$, $B = [4, 6, 8]$, and $T = 13$, then the answer is "yes", since you can score 2 points in frame 1, 3 points in frame 2, and 8 points in frame 3, for a total of 13 points.

   (a) (5 points) For the example above, give a new T value for which the answer is "no".
   $T =$

   (b) (5 points) To show that this problem is NP-Hard, we will reduce from the known NP-Hard problem *Subset-Sum*: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$ and a target integer X, determine whether there is a subset of S that sums to exactly X.

   The reduction works as follows: Given an instance of Subset-Sum with set S and target X, we create an instance of the bowling problem with n frames, arrays A and B, and target T as follows:

   - For each i from 1 to n, set $A[i] = s_i$ and $B[i] = 0$.
   - Set $T = X$.

   Given the the subset-sum instance $S = \{3, 5, 7, 14, 12, 2\}$ and $X = 22$, use the reduction above to create an instance of the bowling problem by filling in the correct values for n, A, B, and T.

   - $n =$
   - $A =$
   - $B =$
   - $T =$

   (c) (10 points) Write a check mark by the true statements below.

   - The reduction above shows that if we can solve the bowling problem in polynomial time, then we can solve Subset-Sum in polynomial time.
   - The reduction above shows that if we can solve Subset-Sum in polynomial time, then we can solve the bowling problem in polynomial time.
   - The reduction above shows that the bowling problem is in NP.
   - The reduction above shows that the bowling problem is NP-Hard.
   - The reduction above shows that the Subset-Sum problem is in NP.
   - The reduction above shows that the Subset-Sum problem is NP-Hard.