



You are taking "[Midterm \(8 hour time limit\)](#)," as a timed exam. [Show more](#)

End My Exam

07:43:17

|            |  |  |  |  |  |  |  |        |
|------------|--|--|--|--|--|--|--|--------|
| < Previous |  |  |  |  |  |  |  | Next > |
|------------|--|--|--|--|--|--|--|--------|

Problem 2

Bookmark this page

Problem 2-1

0/2 points (graded)

Which of the following problems can be solved using dynamic programming? Check all that work.

- ☒ Sum of elements - Given a list of integer elements, find the sum of all the elements.
- ☒ Binary search - Given a list of elements, check if the element X is in the list.
- ☐ Dice throws - Given n dice each with m faces, numbered from 1 to m, find the number of ways to get sum X. X is the summation of values on each face when all the dice are thrown.



Submit

You have used 1 of 1 attempt

✖ Incorrect (0/2 points)

Problem 2-2

2/2 points (graded)

What is the exact probability of rolling at least two 6's when rolling a die three times?

- ☐ 1/12
- ☐ 1/36
- ☒ 2/27
- ☐ 25/27
- ☐ None of the above



Submit

You have used 1 of 1 attempt

✔ Correct (2/2 points)

Problem 2-3

2/2 points (graded)

A greedy optimization algorithm

- ☒ is typically efficient in time.
- ☐ always finds an answer faster than a brute force algorithm.
- ☐ always returns the same answer as the brute force algorithm.
- ☐ never returns the optimal solution to the problem.



Submit

You have used 1 of 1 attempt

✓ Correct (2/2 points)

## Problem 2-4

2/2 points (graded)

Suppose you have a weighted directed graph and want to find a path between nodes A and B with the smallest total weight. Select the most accurate statement.

- ☐ If some edges have negative weights, depth-first search finds a correct solution.
- ☐ If all edges have weight 2, depth-first search guarantees that the first path found to be is the shortest path.
- ☐ If some edges have negative weights, breadth-first search finds a correct solution.
- ☒ If all edges have weight 2, breadth-first search guarantees that the first path found to be is the shortest path.



Submit

You have used 1 of 1 attempt

✓ Correct (2/2 points)

## Problem 2-5

2/2 points (graded)

Which of the following functions are deterministic?

```
import random

def F():
    mylist = []
    r = 1

    if random.random() > 0.99:
        r = random.randint(1, 10)
    for i in range(r):
        random.seed(0)
        if random.randint(1, 10) > 3:
            number = random.randint(1, 10)
            if number not in mylist:
                mylist.append(number)
    print(mylist)

def G():
    random.seed(0)
    mylist = []
    r = 1

    if random.random() > 0.99:
        r = random.randint(1, 10)
    for i in range(r):
        if random.randint(1, 10) > 3:
            number = random.randint(1, 10)
            mvlist.append(number)
```

Calculator

Hide Notes

```
print(mylist)
```

☐ F☐ G☒ Both F and G☐ Neither F nor G

You have used 1 of 1 attempt

✓ Correct (2/2 points)

## Problem 2-6

2/2 points (graded)

Consider a list of positive (there is at least one positive) and negative numbers. You are asked to find the maximum sum of a contiguous subsequence. For example,

- in the list [3, 4, -1, 5, -4], the maximum sum is  $3+4-1+5 = 11$
- in the list [3, 4, -8, 15, -1, 2], the maximum sum is  $15-1+2 = 16$

One algorithm goes through all possible subsequences and compares the sums of each contiguous subsequence with the largest sum it has seen. What is the time complexity of this algorithm in terms of the length of the list,  $N$ ?

☐  $O(1)$ ☐  $O(\log(N))$ ☐  $O(N)$ ☒  $O(N^2)$ ☐  $O(2^N)$ ☐ none of the above

You have used 1 of 1 attempt

✓ Correct (2/2 points)



## edX

- [About](#)
- [Affiliates](#)
- [edX for Business](#)
- [Open edX](#)
- [Careers](#)
- [News](#)

## Legal

- [Terms of Service & Honor Code](#)
- [Privacy Policy](#)
- [Accessibility Policy](#)
- [Trademark Policy](#)
- [Sitemap](#)

## Connect

- [Blog](#)
- [Contact Us](#)
- [Help Center](#)
- [Security](#)
- [Media Kit](#)

