CMPT360

Assignment 2 [80 points] [Sub. Deadline: Canvas]

General Instructions

Please read all the questions carefully. The questions have been designed from scratch. If you find any mathematical error in the question, please report it by email with the subject line 'COMM360'.

There are three types of questions and the rules for collaboration are stated below.

Type X: No discussion on the Canvas discussion board, but you can ask clarification questions over email. The reason this type disallows discussion on Canvas is that they are easy and the slightest hint may reveal the answer. No discussion with any individual within or beyond the class. No online search or AI search or posting.

Type Y: Any clarification question must be asked on the Canvas discussion board. No discussion with any individual within or beyond the class. No online search or AI search or posting.

Type Z: Any clarification question must be asked in the Canvas discussion board. You can discuss this only with the members of your team. Your team can consist of at most 3 students including yourself. No online search or AI search or posting.

Assume all the variables to be integers unless something else is explicitly defined in the context. To receive timely help, it is recommended to ask clarification questions as early as possible. Please see the syllabus for late submission penalties or related policy.

1 Unity is Power [Type X], 15 points

Write an algorithm with the fastest asymptotic running time (to the best of your knowledge) for the following problem, briefly describe why it is a correct algorithm, and analyze the time complexity. If you cannot find any polynomial-time algorithm, then give a backtracking algorithm.

Input: An array $A[1...9n]$ of positive integers, where $n \geq 1$.

Output: We want to change at most $6n$ entries of A such that the array at the end contains exactly two different numbers. If a solution exists then print any solution array after the changes. If no such solution exists then we print 'none'.

Example:

Input: A = [7,1,2,2,3,3,3,5,2]. Output: A = [0,0,2,2,0,0,0,0,2]. Note that here A = [0,0,0,0,3,3,3,0,0] would also be a valid output.

Input: A = [1,9,2,3,7,6,4,8,5]. Output: 'none'

HARD CONSTRAINTS: Your algorithm must be described as a sequence of steps. The use of a pseudocode or a computer program to describe the algorithm is not allowed. Using an operation where an element of A is used as an index of an array is not allowed. The use of a dictionary or any type of hashing is also not allowed.

Submission Instruction: Your answer must provide all three parts: algorithm description, proof of correctness, and time complexity analysis.

Please upload the answers to the questions 1-4 in a single .pdf file.

Marking criteria:

[0 marks] No attempt or entirely unrelated or Wrong Answer or unclear and very hard to read answer

[3 marks] Some related attempt to solve the question

[6 marks] A close to correct answer that followed all submission instructions, but the description and analysis are unclear

[10 marks] A correct answer that followed all submission instructions, and clearly presented the answer, but with slow running time

[15 marks] A correct answer with a fast running time, correct proof and correct time complexity analysis

2 ToH [Type X], 15 points

Write an efficient algorithm to solve the Tower of Hanoi problem with the following modified rule: 'When moving disks, ensure that no smaller disk has more than k larger disks on top of it at any time'. Give a recurrence relation to analyze the running time, and give a closed-form solution of your recurrence relation in terms of n and k, where n is the number of disks.

HARD CONSTRAINTS: Use of computer-assisted solutions or a computer program is not allowed. You can use pseudocode, but explain what it is doing in a description.

Submission Instruction: Your answer must provide all three parts: algorithm description, recurrence relation, and time complexity analysis.

Please upload the answers to the questions 1-4 in a single .pdf file.

Marking criteria:

[0 marks] No attempt or entirely unrelated or Wrong Answer or unclear and very hard to read answer

[3 marks] Some related attempt to solve the question, but very unclear answer, e.g., just produces the algorithm in the book or gives a pseudocode without any relevant explanation

[6 marks] A close to correct answer that followed all submission instructions, but the description and analysis are unclear

[10 marks] A correct answer that followed all submission instructions, and clearly presented the answer, but with a slow running time

[15 marks] A correct answer with a fast running time, correct proof, and correct time complexity analysis

3 Backtrack [Type Y], 15 points

Write a backtracking algorithm to solve the following problem. Give an argument of why the algorithm is correct, and analyze the running time.

Input: An array $A[0...n, 0...n]$ of integers. A robot that starts at position $A[0,0]$ and wants to reach $A[n,n]$. At each step, the robot can only move one step to an adjacent entry (or cell). The cost of a path is the sum of the entries that the robot visits. The number in an entry is the cost for visiting that entry

Output: The most expensive path that the robot can take without revisiting an entry twice.

Example: Input: A =

50 (robot Starts at A[0,0])	*-20*	*300*
-10	*10*	*-70*
110	*40*	*40 (robot ends at A[n,n])*

HARD CONSTRAINTS: Use of computer-assisted solutions or a computer program is not allowed. You can use pseudocode, but explain what it is doing in a description.

Submission Instruction: Your answer must provide all three parts: algorithm description, correctness argument, and time complexity analysis.

Please upload the answers to the questions 1-4 in a single .pdf file.

Marking criteria:

[0 marks] No attempt or entirely unrelated or Wrong Answer or unclear and very hard to read answer

[3 marks] Some related attempts to solve the question, but very unclear answer

[6 marks] A close to correct answer that followed all submission instructions, but the description and analysis are unclear

[10 marks] A correct answer that followed all submission instructions, and clearly presented the answer, but with a slow running time

[15 marks] A correct answer with a fast running time, correct proof, and correct time complexity analysis

4 Lights [Type Y], 15 points

Mr. Minion has arrived in an 2D world, where Mr. Minion is walking from left to right. There are light sources that are above the line at different heights, and the ith light illuminates a triangular area where the pick of the triangle is the light source and the base $[X^\ell[i], X^r[i]]$, where $X^\ell[i] < X^r[i]$, is the line segment it can illuminate. Mr. Minion is given the arrays X^ℓ and X^r. In addition, there is a dictionary L that given $X^\ell[i]$, returns $X^r[i]$ and vice versa. Fortunately, all these numbers are distinct.

Our goal is to find a point p in the line that is illuminated by the maximum number of lights.

Write an algorithm to help Mr. Minion by finding the point p. Write an algorithm, an argument of why the algorithm is correct, and analyze the running time.

HARD CONSTRAINTS: Your algorithm must be described as a sequence of steps. Pseudocodes are not allowed. Mr. Minion does not like to use unnecessary computer memory. So your algorithm can use at most $O(1)$ additional memory. This means you cannot create a new array or binary-search-tree like data structures. If you want to sort, you can use HeapSort which uses $O(1)$ additional memory and takes $O(n \log n)$ time.

Submission Instruction: Your answer must provide all three parts: algorithm description, correctness argument, and time complexity analysis.

Please upload the answers to the questions 1-4 in a single .pdf file.

Marking criteria:

[0 marks] No attempt or entirely unrelated or Wrong Answer or unclear and very hard to read answer

[3 marks] Some related attempts to solve the question, but very unclear answer

[6 marks] A close to correct answer that followed all submission instructions, but the description and analysis are unclear

[10 marks] A correct answer that followed all submission instructions, and clearly presented the answer, but with a slow running time

[15 marks] A correct answer with a fast running time, correct proof, and correct time complexity analysis

5 Image Transfer [Type Z], 20 points

Murphy's law states that 'things will go wrong in any given situation, if you give them a chance,' or more commonly, 'whatever can go wrong, will go wrong.' Many automated software testing tries to test for all possible inputs, whenever the number of possibilities is reasonably small.

An image is normally processed by a set of three satellite modules m_1, m_2, m_3 in this order. The image might get corrupted during the transfer. During this process, a module m_j can request any of its predecessor m_i, where $1 \le i < j \le 3$, to resend the information. Upon receiving a resend request, m_i must pass the data to the next module m_{i+1}.

A normal processing must start at m_1 and end at m_3.

We want to count the number of different ways with exactly n resend requests. If $n = 1$, there are 3 different ways (* denotes the resend request).

```
12*123  [2 gets the corrupt data and decides to ask from 1]
123*23  [3 gets the corrupt data and decides to ask from 2]
123*123 [3 gets the corrupt data and decides to ask from 1]
```

If $n = 2$, then the number of different ways is 8.

```
12*12*123
12*123*123
12*123*23
123*23*23
123*23*123
123*123*23
123*123*123
123*12*123
```

Input:

The first line is the number of test cases. Each test case consists of a number representing n, where $1 \le n \le 20$. For example, in the following input, we have two test cases. The first test case has $n = 1$ and the second has $n = 2$.

Sample input:

```
2
1
2
```

Output:

The output consists of the count representing the number of different ways to transfer an image from m_1 to m_3 with exactly n resend requests.

Sample output:

```
3
8
```

Submission Instruction:

You must edit and submit the given file Image.java

Marking criteria: The code must run without error and pass all our test cases within a minute. No partial mark. If you add/remove/edit the code at any other place except for the suggested area, then the submission will automatically be graded with 0.