COMP4500/7500 Advanced Algorithms and Data Structures - Assignment 2

School: School of Electrical Engineering and Computer Science, The University of Queensland

Semester: 2, 2024

Due Date: 3pm, Friday 18th of October 2024

1. General Information

- This assignment is wor e final grade.
- It is to be attempted in amic programming.

2. Submission

- **Written Answers**: Ans hould be in a pdf file called `A2.pdf`.
- **Source Code**: `Recurs` tted electronically via Blackboard according
- Multiple submissions a one will be saved and marked.
- Submitted work should orrect submissions will receive 0 marks.

3. Late Submission Policy

- A penalty of 10% of the maximum possible mark will be deducted per 24 hours for up to 7 days. After 7 days, a mark of 0 will be given.
- Medical or exceptional circumstances require an extension request via https://my.uq.edu.au/ with a maximum of 7 days from the original deadline.

4. School Policy on Student Misconduct

- Read and understand the School Statement on Misconduct available at https://eecs.uq.edu.au/current-students/student-guidelines/student-conduct. Plagiarism or collusion will result in penalties.

5. Assignment Problem

5.1 Problem Description

- You are in charge of a small microbrewery business for `k` consecutive days. You have a work schedule represented by an array `work` of `k` non-negative integers.
- There are two workers such as maximum number of consecutive days th days off between shifts (`minBreak`), their capa , and salary cost for working on each day (
- A roster for the `k` day of workers scheduled to work on that day. A ros .
- The goal is to find a va l cost.

5.2 Example

- Consider `k = 5` days,
  `w0 = (maxShift = 2,` ost = [1,2,2,1,1])` , and
  `w1 = (maxShift = 1,` st = [0,1,1,1,2])` .
- An optimal roster is `roster = [{ w0 , w1 },{},{ w0 },{},{ w0 }]` with a total cost of 33.

5.3 Tasks

1. **(a) Optimal Substructure - Recursive Solution (20 marks)**
   - Implement the `public static method optimalRecursive` in the `Recursive` class to provide a naive recursive algorithm to determine the total cost of an optimal valid roster. The method should not return the roster itself but only the total cost.
2. **(b) Time Complexity of Recursive Algorithm (15 marks)**
   - Give an asymptotic lower bound on the worst-case time complexity of the recursive algorithm in terms of `k` . Provide a lower-bound recurrence, justify it, and solve it.
3. **(c) Dynamic Programming Solution (30 marks)**