CSE204: Complexity of Algorithms

Coursework Assessment 3

Department of Computer Science & Software
Engineering Xi'an Jiaotong-Liverpool University

Assessment Information

Assessment Number	3
Contribution to Overall Mark	15%
Submission deadline	12 June

Assessment Objectives

This assessment aims to test the followings:

- Your fluency on utilising basic number theory applications, e.g., cryptography.
- Your fluency on formal theories providing evidence that many important computational problems are inherently intractable, e.g., reduction.

Instructions

- **This is an INDIVIDUAL coursework. It is recommended to write your program in JAVA.**
- **Write a short report (up to 5 pages) to describe your code, show the results, and answer the questions.**
- **The implementation code and your report should be submitted as a soft copy on ICE. Only the ".zip" file type is acceptable.**
- **Late submissions will be penalised according to the XJTLU University rules.**

1. Write a program that uses a pseudorandom one-time pad.

1) Rather making a truly **random one-time pad**, make the program use a pseudorandom number generator to make a pseudorandom pad when it starts. As you encrypt and decrypt messages, make the program keep track of the characters in one-time pad that have been used for encryption and decryption. The detailed requirements are shown as follows: [40 marks]

   a. Write a function to generate a truly random one-time pad use a pseudorandom number generator. The pseudorandom number generator can be a build-in function in JAVA. The one-time pad must be different at every time you start the program.
   b. Write a function to encrypt the message to ciphertext and a function to decrypt the ciphertext to message.
   c. The program can be a console-based program or a window-based program with a graphical user interface.
   d. The program accepts the input strings from the keyboard. Only English characters and numbers are acceptable. The case of English characters and "space" characters must be ignored in the encryption and decryption. E.g. the input string is "This is a Message". Before the encryption, this message must reduce to "thisisamessage", and the output of the decryption also be "thisisamessage". The original message, reduced message, ciphertext, and the output of the decryption must be clearly shown on the screen.
   e. When the program runs the encryption or decryption function, it must show what characters in the one-time pad are using, what characters in the one-time pad have been used, and what characters in the one-time pad are not used.

2) Explain how a cryptographically secure random number generator is equivalent to an unbreakable encryption scheme. In other words, if you have a cryptographically secure random number generator, how could you use it to create an unbreakable encryption scheme and vice versa? (Answer this question) [20 marks]

2. Suppose you are given a set of objects with weights Wi and values Vi, and a knapsack that can hold a total weight of W. Then three forms of the knapsack problem are as follows:

    - Detection—For a value k, is there a subset of objects that fit into the knapsack and have a total value of at least k?
    - Reporting—For a value k, find a subset of objects that fit into the knapsack and have a total value of at least k if such a subset exists.
    - Optimisation—Find a subset that fits in the knapsack with the largest possible total value.

    Suppose that the function of Detection is given, which is **DetectKnapsack(k)**. This function returns **TRUE** when there is a subset of objects that fit into the knapsack and have a total value of at least k, and it returns **FALSE** when there is no subset of objects that fit into the knapsack and have a total value of at least k.

    1). Write a **Pseudocode Function** to find a reduction of the **reporting** problem to the **detection** problem. [20 marks]

    2). Write a **Pseudocode Function** to find a reduction of the **optimisation** problem to the **detection** problem. [20 marks]

    Hint:

    - ***This question is not asking you to write a program but only a Pseudocode.***
    - ***This question is an Open book Question. You should learn what is reduction and how to reduce a problem to another problem (See Appendix).***

Appendix

Detection, Reporting, and Optimization Problems

Many interesting problems come in three forms: detection, reporting, and optimization. The detection problem asks if a solution of a given quality exists. The reporting problem asks you to find a solution of a given quality. The optimization problem asks you to find the best possible solution.

For example, in the subset sum problem, you are given a set of numbers, and there are three associated problems:

- Detection—Is there a subset of the numbers that adds up to a specific value k?
- Reporting—Find a subset of the numbers that adds up to the specific value k, if such a subset exists.
- Optimization—Find a subset of the numbers with a total as close to the specific value k as possible.

(A variation on the subset sum problem asks you to find a subset with values that total 0.)

At first some of these problems may seem easier than others. For example, the detection problem only asks you to prove that a subset adds up to 0. Because it doesn't make you find subsets like the reporting problem does, you might think the detection problem is easier. In fact, you can use reductions to show that the three forms of problems have the same difficulty, at least as far as complexity theory is concerned. To do that, you need to show four reductions:

- Detection ≤p Reporting
- Reporting ≤p Optimization
- Reporting ≤p Detection
- Optimization ≤p Reporting

Reductions are transitive, so the first two reductions show that Detection ≤p Reporting ≤p Optimization, and the second two reductions show that Optimization ≤p Reporting ≤p Detection.