Programming Assignment 2 (DRAFT)

COMP526: Efficient Algorithms
Fall 2024

DUE Friday, 29 November by 11:59pm

Background

Motivation

COMP666: Infernal Computing has a devilish final exam. The exam consists of 20 true/false questions that are worded in such a confusing way that you are better off guessing each answer rather than trying to understand the question. Given that you need to pass the exam in order graduate, you decide to cheat. You and your leet hacker friend come with an elaborate scheme to pull it off.

You learn that the professor uses a program ot auto-generate exam questions and solutions a few minutes before the exam. Using her leet hacking skills, your friend can get access to the answers once they are generated. This gives her just a minute to secretly pass the solutions to you. The two of you devise a system where she stands outside the classroom where the exam takes place and coughs in such a way to encode information about the exam solutions: a short cough for "false" and a long cough for "true." Unfortunately, she can only reliably send 10 answers before raising suspicions, while the exam consists of 20 true/false questions. Given that your friend will know all 20 answers but can only relay 10 true/false values, you decide to come up with a coding scheme to maximize your *worst case* score on the exam with the information your friend can send you. What is the highest guaranteed score you can achieve with this scheme?

Formalizing the Problem

Formally, the (correct) exam answers consist of 20 true/false responses. We can represent these responses as a binary array $B[0..20)$ where $B[i] = 1$ if the (correct) answer to question i is true, and $B[i] = 0$ if the (correct) answer to question i is false.

Given the array B, your friend must determine a message M that she sends to you. We can similarly encode $M = M[0..9)$ as a binary array containing only 10 values. Given the message M, you must determine your corresponding answers to the questions which we can represent as a binary array of length 20 $A[0..19)$. The challenge is to determine an encoding scheme:

- An encoding function $e : \{0,1\}^{20} \to \{0,1\}^{10}$ from which your friend computes her message from the correct answers. That is, $M = e(B)$.

- A decoding function $d : \{0,1\}^{10} \to \{0,1\}^{20}$ from which you determine your exam answers from your friend's message. That is, $A = d(M)$.

Given exam solutions B and your answers A, your score on the exam is $\text{score}(B, A) = 20 - H(A, B)$, where

$$H(A, B) = |\{i \mid A[i] \neq B[i]\}|$$

is the *Hamming distance* between A and B.

Your goal is to define encoding and decoding functions e and d that maximize your guaranteed score. That is, you wish to maximize the value

$$G = \min\left\{\text{score}(B, d(e(B))) \,\middle|\, B \in \{0,1\}^{20}\right\}.$$

Note that this expression is the worst-case (i.e., lowest) score that your encoding scheme provides over all possible exam solutions.

Assignment Description

For this assignment, you will implement an encoding scheme as described above in the Python programming language. To this end, you must complete the program `exam_cheat_code.py` by implementing two functions:

- `encode(solutions: list[int]) -> list[int]` that takes as input a list of 20 binary (0-1) values representing the correct solutions to the exam (B), and returns a list of 10 binary values representing the message M to be relayed to you during the exam.
- `decode(message: list[int]) -> list[int]` that takes as input a list of 10 binary values representing the message M you receive from your friend, and returns a list of 20 binary values representing the solutions A you submit for the exam.

Your completed `exam_cheat_code.py` will be tested using the program `exam_tester.py`. This program iterates over all 2^{20} possible exam solutions and finds the lowest score guaranteed by your implementation of the encoding and decoding functions.

In addition to the completed `exam_cheat_code.py`, you must submit a brief (~ 1 page) typed report on how your encoding/decoding scheme works. Your report should explain any systematic approaches you took, and any extra code you wrote/used to find your final encoding scheme.

Optional. You may also include a write-up of impossibility results. To this end, you should prove an upper bound on the guaranteed score from any encoding scheme that uses a message size of $|M| = 10$ bits.

What to Submit

Assignment submission is through Canvas. For your submission, you should upload the following files. Please ensure that you name the files ***exactly*** as stated below:

- `source.zip` a ZIP archive containing your implementation of `exam_cheat_code.py` and any other code you wrote for designing and testing your scheme.
- `explanation.pdf` a single page typed PDF file containing your write-up and explanation of your program.
- (*Optional*) `impossibility.pdf` a typed PDF file containing your argument or explanation of an impossibility result showing an upper bound on the maximum achievable guaranteed score for any encoding scheme.

Resources

If you are new to Python (or want a refresh on the basics) you may find the following website helpful:

- `https://www.pythoncheatsheet.org`

Marking

Your submission will be marked on a scale from 0 to 100. The breakdown of marking will be as follows:

- 70 marks for code. The higher your guaranteed exam score, the more marks. For submissions giving the same guaranteed exam score, more marks will be awarded to the *simpler* submission. Submissions with a guaranteed score of more than 16 may receive more than 70 marks in this category.
- 30 marks for explanation. Full marks will be awarded to concise and clear explanations that are consistent with the provided code.
- Up to 20 marks of extra credit may be awarded for a write-up of impossibility results included with your submission.