Project 1

Information theory

Each student should (individually) write two programs in Python 3, using the `bitstring` library, in order to compress and decompress files with the Lempel–Ziv algorithm that was taught in class.

- The program `compress.py` should accept two arguments: the name of the file to be compressed, and the name of the output file where the result of compression should be stored. Hence the command:

  ```
  $ python compress.py file.txt file.lz
  ```

  assumes that a file named `file.txt` exists, and creates, or overwrites, a file called `file.lz`

- The program `decompress.py` should accept two arguments, giving the name of the file to be decompressed, and the name of the file where one stores the result of decompression. Hence, the command

  ```
  $ python decompress.py file.lz file-2.txt
  ```

  assumes that there exists a file `file.lz` and creates or overwrites a file `file-2.txt`.

- Naturally, the commmand

  ```
  $ diff file.txt file-2.txt
  ```

  should report that the two files are identical. (Make sure that this works and test with numerous examples.)

Evaluation. After submissions I will test the programs and inspect the code. Possibly I will do an oral interview (maybe in all cases, maybe in some cases only).

- If the program is correct and I'm fully convinced that the student actually did the code him/herself, then the student will be given 4 points (which will contribute to the sum-total of the student's grade).
- If the program is not correct, but I am convinced that the student wrote it, then the student will be given 0 points.
- If I am not convinced that the student wrote the program by him/herself, then the student will fail the course and will be reported to the faculty ethics board.

- The submission should consist of a single zip file, with the filename:

  `<student ID number>-<full name>.zip`

which should contain two files `compress.py` and `decompress.py`, and possibly a third file `readme.txt` (optional). And NO OTHER FILE.

- The scripts will be executed automatically. The scripts should not read any file other than the ones passed as argument, or access the internet, etc.

- Otherwise, the implementations should be reasonably efficient. As an upper limit, compression of a 500 Kb file should take less than 10 minutes[1] and use less than 300 Mb of RAM. The program will be aborted otherwise. Note that this limit is way beyond what would be reasonably for an actual tool.

  If the algorithm takes longer or occupies too much memory, it will be considered wrongly implemented.

[1] In CPython. Less than 2 minutes in PyPy.