

Απαλλακτική Εργασία στο Μάθημα: ΘΕΩΡΙΑ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΙ ΚΩΔΙΚΩΝ

Έτος: 2024

Εξάμηνο: 4^ο

Ονοματεπώνυμο: Χρήστος Λαζαρίδης (p22083), Λουΐζος Βασίλειος (p22252) , Φετί Οσμά (p22126)

Υλοποίηση: $(22083 + 22252 + 22126) \bmod 9 = 6$, Compression: Fano-Shannon, encoding: Linear σε python με τις βιβλιοθήκες Flask και Numpy

Όλα τα αρχεία source code των κωδίκων βρίσκονται στο φάκελο codes. Για την υλοποίηση

- Fano-Shannon: Για την υλοποίηση έγινε η κλάση “Compression” που χρησιμοποιείτε για compressions/decompression ανάλογα με τη μεταβλητή mode του constructor. Για το compression, αρχικά υπολογίζει τη συχνότητα του κάθε χαρακτήρα, φτιάχνει το code_table (το οποίο στη περίπτωση του decompression πρέπει να περαστεί ως παράμετρος) και κωδικοποιεί με βάση αυτών το κείμενο που του δίνεται. Για την αποκωδικοποίηση, απλά αποκωδικοποιεί ένα αρχείο bit, σε κείμενο με βάση το πίνακα χαρακτήρων. Ο πίνακας αυτός περνιέται στο server ως παράμετρος.
- Γραμμικός (linear) Κώδικας: Η υλοποίηση επιτρέπει στο χρήστη να επιλέξει τον αριθμό των bit της κωδικοποιημένης λέξης ενώ το κάθε μήνημα κωδικοποιείτε σε 4αδες. Στη Κλάση LinearCode, αρχικά υπολογίζονται η μήτρα ισοτιμίας(P) και ο γεννήτορας πίνακας(G), κατά τα γνωστά της θεωρίας. Η μήτρα ισοτιμίας υπολογίζεται κάνοντας πράξεις με τις θέσεις i και j για κάθε συγκεκριμένη θέση. Συγκεκριμένα, τα bit του j + 1 (λόγω του μηδενικού index της python) θα γίνουν αριστερό δεξί κατά i θέσεις έτσι ώστε να απομωνώσουμε τα ενδιαφέροντα bit στις least significant θέσεις και το αποτέλεσμα γίνεται λογικό end με το 1, αυτές οι πράξεις εξασφαλίζουν το ότι δεν θα υπάρχουν 2 ίδιες γραμμές και πως για 2 παραμέτρους n, k ο πίνακας θα είναι πάντα ο ίδιος. Έπειτα το μήνημα γίνεται padded έτσι ώστε να είναι πολλαπλάσιο του 4 και προστίθεται ένα header 2 bit (max padding = 3 bits). Έπειτα κωδικοποιείτε το μήνημα. Για την αποκωδικοποίηση, υπολογίζεται κάθε λέξη του πίνακα και ο πίνακας Checker, που περιέχει στη πρώτη στήλη, όλα τα πιθανά σύνδρομα (αποθηκευμένα σε μορφή integer) και στη πρώτη γραμμή όλες τις πιθανές λέξεις του κώδικα, στη συνέχεια για κάθε πιθανή λέξη υπολογίζουμε το σύνδρομο, στη περίπτωση αλλοίωσης ενός bit τη φορά, σε κάθε μια από τις θέσεις της και το αποτέλεσμα της αλλοίωσης τοποθετείτε στη στήλη της αρχικής λέξης και τη γραμμή του συνδρόμου που υπολογίζεται με τη αλλοιωμένη λέξη και το πίνακα H (ο πίνακας H υπολογίζεται από τη θεωρία κατά τα γνωστά). Αποτέλεσμα είναι πως εφόσον η αλλοιωμένη λέξη βρεθεί στο checker matrix, η εύρεση του συνδρόμου της θα μας οδηγήσει κατευθείαν στη αρχική. Η αποκωδικοποίηση γίνεται σε n-αδες και στο τέλος αφαιρείτε το padding. Μετά τη πιθανή διόρθωση της λέξης, απλά κρατάμε τα k αρχικά bits. Ο κώδικας μπορεί να διορθώσει κάθε λέξη με το πολύ ένα λάθος ανα n-αδα αλλά όπως θα διαπιστώσετε με περισσότερα λάθη ανά n-αδα, που τήνει να συμβαίνει με τυχαία συνάρτηση

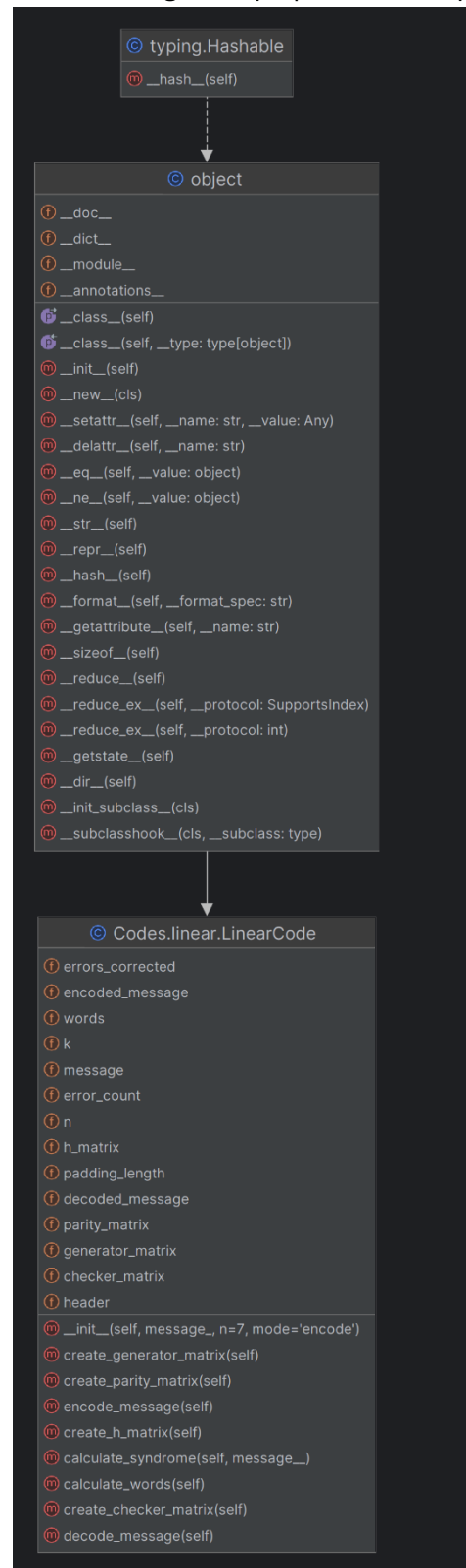
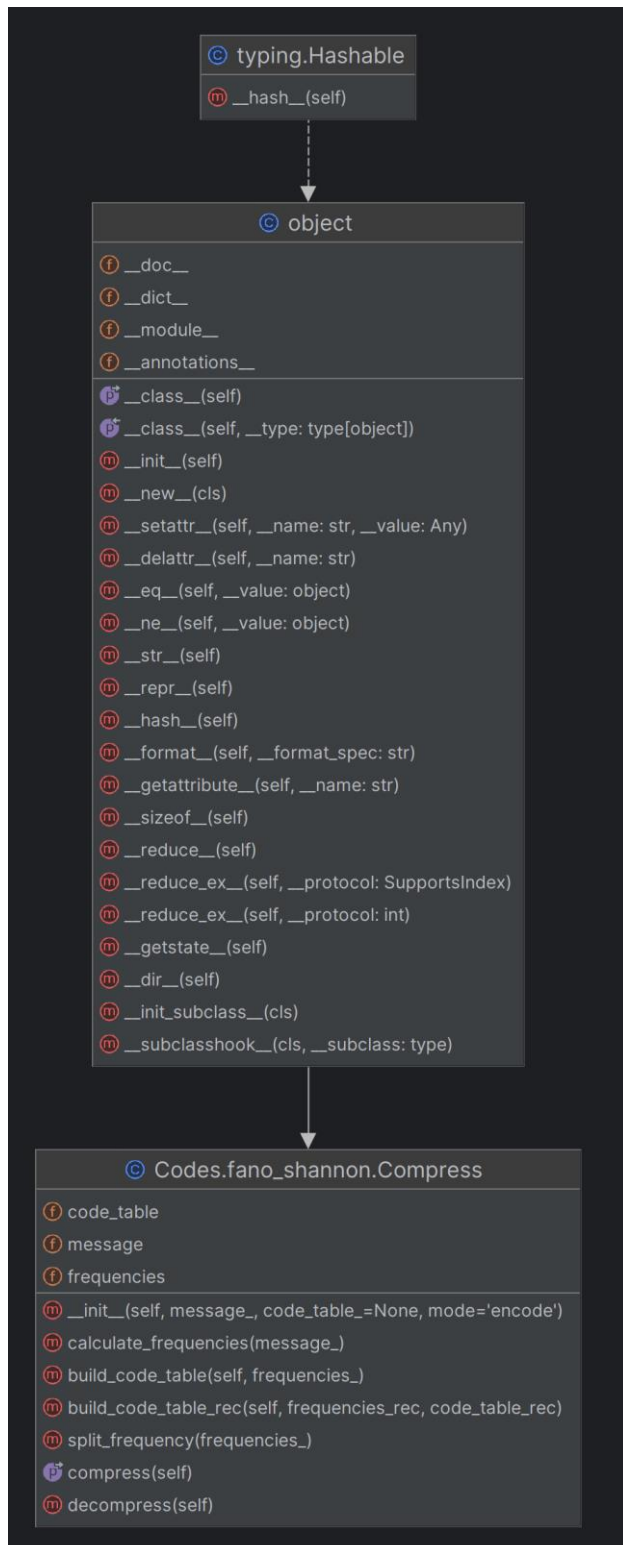
θορύβου, τότε η διόρθωση σφαλμάτων δεν είναι αποδοτική, ειδικά σε ποσοστά άνω του 5%, αυτό συμβαίνει εξαιτίας της φύσης το απλού γραμμικού κώδικα

Διαγράμματα κλάσεων:

Για τον κώδικα Fano-Shannon:

Για τον κώδικα Linear:

Τα δεδομένα τα οποία εισάγονται από τον Client είναι “message”, ο αριθμός των bit της



κωδικοποιημένης λέξης το γραμμικού κώδικα και το ποσοστό του θορύβου. Στο server στέλνεται μετά από τις απαραίτητες κωδικοποιήσεις, το κωδικοποιημένο μήνυμα σε (κατά σειρά κωδικοποίησης) fano-shannon -> linear code -> base64, οι κώδικες που χρησιμοποιήθηκαν, το code table του fano-shannon, τα bit της κωδικοποιημένης λέξης του γραμμικού κώδικα, τα συνολικά λάθη και η εντροπία. Τα λάθη υπολογίζονται με τελείως τυχαίο τρόπο, με βάση το ποσοστό (χρήση της βιβλιοθήκης random της python). Η λογική αυτή βρίσκεται στο script client.py το οποίο και θα πρέπει να τρέξει ο χρήστης, την ώρα που το αρχείο app.py που περιέχει το flask server επίσης τρέχει στο url: <http://127.0.0.1:5000>. Ο server θα αποκωδικοποιήσει το κείμενο με την αντίστροφη σειρά κωδικοποίησης του (base64 -> linear -> fano-shannon) και θα απαντήσει με ένα json που θα περιέχει το αποκωδικοποιημένο μήνυμα, τον αλγόριθμο συμπίεσης, τον κώδικα (όπως τα πήρε από το client.py και εδώ θα είναι πάντα fano-shannon και linear αντίστοιχα), τα λάθη τα οποία υπήρχαν, τα λάθη που διορθώθηκαν μέσω της χρήσης γραμμικού κώδικα και την εντροπία του τελικού μηνύματος (και τη εντροπία του αρχικού όπως την υπολόγισε το client.py).

Στον κώδικα του client.py γίνεται κωδικοποίηση των δεδομένων από τον χρήστη και στην συνέχεια αποστέλλουμε τα δεδομένα στον server, ο οποίος τα αποκωδικοποιεί.