Πανεπιστήμιο Ιωαννίνων Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Αλγόριθμοι και Πολυπλοκότητα

3η Εργασία

Επίλυση προβλήματος μέγιστης κοινής υποακολουθίας με δύο αλγορίθμους

Όνομα: Δημήτριος

Επώνυμο: Κάτσανος

ΑΜ: 2281

Email: int02281@uoi.gr

Πίνακας Περιεχομένων

1. Υλικό που χρησιμοποιήθηκε

Το υλικό που χρησιμοποιήθηκε για τα πειράματα είναι:

- Τύπος συστήματος: Λειτουργικό σύστημα 64 bit, επεξεργαστής τεχνολογίας x64
- Επεξεργαστής: Intel Core i5 2.70GHz
- Εγκατεστημένη RAM:8,00GB DDR3 (798MHz)

2. Λογισμικό που χρησιμοποιήθηκε

Το λογισμικό που χρησιμοποιήθηκε είναι:

- Python 3.8.10
- Visual Studio Code

3. Περίληψη

Η επίλυση του προβλήματος της μέγιστης κοινής υποακολουθίας έχει πολλές εφαρμογές. Η συγκεκριμένη εργασία ζητείτε η επίλυση του προβλήματος της μέγιστης κοινή υποακολουθίας DNA. Η επίλυση γίνεται με δύο διαφορετικούς αλγορίθμους:

- Τον αλγόριθμο brute force
- Τον αλγόριθμο longest Common Subsequence

Οι ακολουθίες DNA αναπαρίστανται με συμβολοσειρές που σχηματίζονται από 4 χαρακτήρες(A,G,C,T) που αναπαριστούν τα νουκλεοτίδια αδενίνη, γουανίνη, κυτοσίνη και θυμίνη.

4. Εισαγωγή

Στο πρόβλημα της μέγιστης κοινής υποακολουθίας του DNA έχουμε δύο συμβολοσειρές DNA με μήκος m η μία και n η άλλη και πρέπει να βρούμε την μέγιστη κοινή υποακολουθία αυτών των δύο. Η επίλυση του προβλήματος επιτυγχάνεται με τον απλοϊκο αλγόριθμο ωμής δύναμης (brute force) και τον αλγόριθμο μέγιστης κοινής υποακολουθίας. Ο πρώτος αλγόριθμος δημιουργεί όλες τις υποακολουθίες της πρώτης ακολουθίας (2^m σε πλήθος υποακολουθίες) και ελέγχει ποια είναι η μεγαλύτερη κοινή ακολουθία με την δεύτερη. Ο δεύτερος είναι ο αλγόριθμος δυναμικού προγραμματισμού, του οποίου μας δίνεται στην εκφώνηση ο ψευδοκώδικας

5. Αποτελέσματα

Στην εργασία χρησιμοποιήθηκαν οι δυο αλγόριθμοι LCS και brute force

Η εκφώνηση της εργασίας μας έλεγε να τρέξουμε ένα παράδειγμα με 1000 υποθετικές ακολουθίες DNA με 2000 χαρακτήρες η κάθε μια. Η εργασία μου τρέχει για υποθετικές ακολουθίες DNA με 5 χαρακτήρες

Τα αποτελέσματα από τους δύο αλγορίθμους εμφανίζονται παρακάτω:

```
C:\Users\user\Documents\ΚΑΤΣΑΝΟΣ_2281_3ΕΡΓΑΣΙΑ>main.py

The results from brute force algorithm are
 [('GTCTACAATA', 'GTCTACAATA', 'GTCTACAATA', 10), ('ACGAAGACAC', 'ACGAAGACAC', 'ACGAAGACAC', 10), ('TTTGTTGCGA',
GCGA', 'TTTGTTGCGA', 10), ('GAGTCTTATG', 'GAGTCTTATG', 'GAGTCTTATG', 10), ('ACGAAGACAC', 'ACGAAGACAC', 'ACGAAGACA
, ('TTTGTTGCGA', 'TTTGTTGCGA', 'TTTGTTGCGA', 10), ('GAGTCTTATG', 'GAGTCTTATG', 'GAGTCTTATG', 10), ('TTTGTTGCGA',
GCGA', 'TTTGTTGCGA', 10), ('GAGTCTTATG', 'GAGTCTTATG', 'GAGTCTTATG', 10), ('GAGTCTTATG', 'GAGTCTTATG', 'GAGTCTTAT
]

The results from  algorithm are Longest Common Subsequence
 [('TATTCGACGA', 'TTTGTTGCGA', 'TTTGCGA', 7)]
```

Οι εξής εντολές έδωσα για να εμφανίσει τα αποτελέσματα

Για τον main κώδικα χρησιμοποιούμε την εντολή: main.py

Για τα unit tests χρησιμοποιούμε την εντολή: Unittest.py

6. Συμπεράσματα

Μέσα από αυτή την εργασία κατανοούμε τον τρόπο που μπορούμε να εντοπίσουμε την μέγιστη κοινή υποακολουθία, καθώς και την διαφορά των δύο αλγορίθμων οπού ο πρώτος τρέχει πιο αργά από τον δεύτερο αφού ο πρώτος έχει πολυπλοκότητα O(2^n) και ο δεύτερος έχει O(mn)