Πανεπιστήμιο Ιωαννίνων Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Αλγόριθμοι και Πολυπλοκότητα

Επίλυση προβλήματος μέγιστης κοινής υποακολουθίας με δύο αλγορίθμους

Σταυροπούλου Μαρία

ΑΜ: 2058

30 Ιανουαρίου 2022

Περίληψη

Η επίλυση του προβλήματος της μέγιστης κοινής υποακολουθίας έχει πολλές εφαρμογές. Στην συγκεκριμένη εργασία ζητείτε η επίλυση του προβλήματος της μέγιστης κοινής υποακολουθίας DNA. Συγκεκριμένα 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.

1 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 απλοϊκό 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 τη δεύτερη. Ο δεύτερος είναι αλγόριθμος δυναμικού προγραμματισμού, του οποίου μας δίνεται στην εκφώνηση ο ψευδοκώδικας.

2 Αποτελέσματα

Για την συγγραφή του κώδικα χρησιμοποιήθηκε η Python 3.9.6 και το Visual Studio Code. Τα χαρακτηριστικά του υπολογιστή είναι i3-4000M(2.40GHz), 4 GB DDR3.

Ο αλγόριθμος LCS υλοποιείτε με τον εξής τρόπο:

- Αρχικά δεχόμαστε τις ακολουθίες και βρίσκουμε το μήκος τους (m,n)

- Έπειτα δημιουργούμε έναν πίνακα διαστάσεων n+1*m+1 όπου την πρώτη σειρά και την πρώτη στήλη γεμίζουν με μηδενικά.
- Συμπληρώνουμε κάθε κελί του πίνακα χρησιμοποιώντας την παρακάτω λογική.
- Εάν ο χαρακτήρας που αντιστοιχεί στην τρέχουσα σειρά και την τρέχουσα στήλη ταιριάζουν, τότε συμπληρώνουμε το τρέχον κελί προσθέτοντας ένα στο διαγώνιο στοιχείο.
- Διαφορετικά, πάρτε τη μέγιστη τιμή από την προηγούμενη στήλη και το στοιχείο της προηγούμενης γραμμής για τη συμπλήρωση του τρέχοντος κελιού. Τοποθετήστε ένα βέλος στο κελί με τη μέγιστη τιμή. Αν είναι ίσα, υποδείξτε κάποιο από αυτά.
- Η τιμή στην τελευταία σειρά και στην τελευταία στήλη είναι το μήκος της μεγαλύτερης κοινής υποακολουθίας.
- Για να βρούμε και το ποια είναι αυτή η κοινή υποακολουθία ξεκινάμε από το τελευταίο στοιχείο και πηγαίνουμε προς τα πίσω, μέχρι να συναντήσουμε την πρώτη θέση που παρουσιάζεται ο προηγούμενος αριθμός

Ο αλγόριθμος αυτός τρέχει πολύ γρηγορότερα από ότι ο αλγόριθμος ωμής δύναμης.

Στην εκφώνηση της εργασίας μας έλεγε να τρέξουμε ένα παράδειγμα με 1000 υποθετικές ακολουθίες DNA με 2000 χαρακτήρες η κάθε μια. Όμως επειδή πήγε να κολλήσει ο υπολογιστής τελικά το έτρεξα με μικρότερο αριθμό(10 υποθετικές ακολουθίες DNA με 5 χαρακτήρες η κάθε μια).

Τα αποτελέσματα από τους δύο αλγορίθμους εμφανίζονται παρακάτω.

```
The results from brute force algorithm are
 [('TACTT', 'AACTT', 'ACTT', 4), ('GACAC', 'CGAAC', 'GAAC', 4), ('TCTAA', 'TGTAA', 'TTAA', 4)]

The results from  algorithm are Longest Common Subsequence
 [('TACTT', 'AACTT', 'ACTT', 4), ('GACAC', 'CGAAC', 'GAAC', 4), ('TCTAA', 'TGTAA', 'TTAA', 4)]
```

Εικόνα 1: Παράδειγμα αποτελεσμάτων που εμφανίζονται στο χρήστη.

3 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)