Τρίτη εργασία στο μάθημα "Αλγόριθμοι και Πολυπλοκότητα"

Γκόγκος Χρήστος
Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

Άρτα, Ιανουάριος 2022

1 Εισαγωγή

Ζητείται η επίλυση του προβλήματος της μέγιστης κοινής υποακολουθίας [Bla20], [Sta98]. Η επίλυση του προβλήματος, μεταξύ άλλων, έχει εφαρμογή στη βιοπληροφορική, και ειδικότερα στην ανάλυση ομοιοτήτων σε ακολουθίες DNA. Οι ακολουθίες DNA αναπαρίστανται με συμβολοσειρές που σχηματίζονται από 4 χαρακτήρες (A,G,C,T) που αναπαριστούν τα νουκλεοτίδια αδενίνη, γουανίνη, κυτοσίνη και θυμίνη.

2 Μέγιστη κοινή υποακολουθία (LCS=Longest Common Subsequence)

Έστω ότι δίνονται δύο συμβολοσειρές `X` και `Y` με μήκη `m` και `n` αντίστοιχα. Στο πρόβλημα της μέγιστης κοινής υποακολουθίας ζητείται να βρεθεί η μέγιστη σε μήκος υποακολουθία που εντοπίζεται και στις δύο συμβολοσειρές `X` και `Y`. Η υποακολουθία αποτελείται από χαρακτήρες που μπορούν να εντοπιστούν και στις δύο συμβολοσειρές με την ίδια σειρά από αριστερά προς τα δεξιά. Για παράδειγμα αν `X="AATCGAG"` και `Y="CCATCGG"` τότε η μέγιστη κοινή υποακολουθία είναι η `"ATCGG"` με μήκος 5.

Γράψτε έναν απλοϊκό αλγόριθμο ωμής δύναμης (brute force) για την επίλυση του προβλήματος. Ο αλγόριθμος αυτός να δημιουργεί όλες τις υποακολουθίες του `X` (2^m σε πλήθος υποακολουθίες) και να ελέγχει ποια είναι η μεγαλύτερη που υπάρχει και στο `Y`. Στη συνέχεια υλοποιήστε αλγόριθμο δυναμικού προγραμματισμού, ο ψευδοκώδικας του οποίου δίνεται στον Αλγόριθμο 1. Επεκτείνετε τον αλγόριθμο έτσι ώστε να επιστρέφει εκτός από το μήκος της μέγιστης υποακολουθίας και την ίδια τη μέγιστη υποακολουθία. Μια χρήσιμη οπτικοποίηση του αλγορίθμου εντοπίζεται στο https://www.cs.usfca.edu/ galles/visualization/DPLCS.html. Ο πίνακας απομνημόνευσης αποτελεσμάτων των δευτερευόντων προβλημάτων για `X="AATCGAG"` και `Y="CCATCGG"` παρουσιάζεται στο Σχήμα 1.

Algorithm 1 Αλγόριθμος μέγιστης κοινής υποακολουθίας (υπολογισμός μήκους)

```
procedure LCSubsequence(X, Y)
    m ← length(X)        ▷ m είναι το μήκος της συμβολοσειράς X
    n ← length(Y)        ▷ n είναι το μήκος της συμβολοσειράς Y
    for i ← 1 to m do
        c[i, 0] ← 0
    end for
    for j ← 1 to n do
        c[0, j] ← 0
    end for
    for i ← 1 to m do
        for j ← 1 to n do
            if X[i] == X[j] then
                c[i, j] ← c[i − 1, j − 1] + 1
            else
                c[i, j] ← max(c[i − 1, j], c[i, j − 1])
            end if
        end for
    end for
    Επέστρεψε το c[m, n]
end procedure
```

			A	A	T	C	G	A	G
		-1	0	1	2	3	4	5	6
	-1	0	0	0	0	0	0	0	0
C	0	0	0	0	0	1	1	1	1
C	1	0	0	0	0	1	1	1	1
A	2	0	1	1	1	1	1	2	2
T	3	0	1	1	2	2	2	2	2
C	4	0	1	1	2	3	3	3	3
G	5	0	1	1	2	3	4	4	4
G	6	0	1	1	2	3	4	4	5

Σχήμα 1: Ο πίνακας απομνημόνευσης αποτελεσμάτων δευτερευόντων προβλημάτων για X=AATCGAG και Y=CCATCGG για τον αλγόριθμο δυναμικού προγραμματισμού.

3 Πειράματα

Δημιουργήστε με τυχαίο τρόπο 1000 υποθετικές ακολουθίες DNA με 2000 χαρακτήρες η κάθε μια. Εντοπίστε όλες τις ακολουθίες DNA που έχουν τη μεγαλύτερη ομοιότητα. Εκτυπώστε τις ακολουθίες αυτές.

4 Παραδοτέα εργασίας

Τα παραδοτέα της εργασίας είναι τα ακόλουθα:

1. Κώδικας που υλοποιεί τους 2 αλγορίθμους που ζητούνται (ωμής δύναμης και δυναμικού προγραμματισμού).
2. Unit tests ελέγχου της ορθότητας των αλγορίθμων.
3. Οδηγίες εκτέλεσης του κώδικα και των unit tests.
4. Τεχνική αναφορά για την εργασία, στα πρότυπα σύντομου επιστημονικού άρθρου. Η αναφορά θα πρέπει να είναι περίπου 2 σελίδες και να περιέχει:
   (α′) Τα χαρακτηριστικά του υπολογιστή και του λογισμικού που χρησιμοποιήθηκε στα πειράματα.
   (β′) Σχολιασμό σχετικά με τον αλγόριθμο Longest Common Subsequence.
   (γ′) Συνοπτική παρουσίαση των αποτελεσμάτων από την εκτέλεση του πειράματος.

5 Παρατηρήσεις

- Η υλοποίηση του κώδικα θα πρέπει να γίνει κατά προτίμηση στη γλώσσα προγραμματισμού Python.
- Η εργασία είναι ατομική και η παράδοσή της γίνεται στο ecourse του μαθήματος μέχρι τις 23/01/2022.

Αναφορές

[Bla20] Paul E. Black. Longest Common Subsequence. Algorithms and Theory of Computation Handbook, CRC Press LLC, 1999 `https://xlinux.nist.gov/dads/HTML/longestCommonSubsequence.html`, 2020. [Online; accessed 2-January-2022].

[Sta98] John Stasko. CS 3158 - Design and Analysis of Algorithms: Longest Common Subsequence. `https://www.cc.gatech.edu/classes/cs3158_98_fall/lcs.html`, 1998. [Online; accessed 2-January-2022].