Τρίτη εργασία στο μάθημα "Αλγόριθμοι και Προχωρημένες Δομές Δεδομένων"

Μεταπτυχιακό Πληροφορικής και Δικτύων

Γκόγκος Χρήστος
Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

Άρτα, Ιανουάριος 2022

1 Εισαγωγή

Ζητείται η επίλυση του προβλήματος της μέγιστης κοινής υποσυμβολοσειράς [Bla20b] καθώς και η επίλυση του προβλήματος της μέγιστης κοινής υποακολουθίας [Bla20a], [Bha16]. Οι αλγόριθμοι αυτοί, μεταξύ άλλων, έχουν 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 Περιγραφή προβλημάτων

2.1 Μέγιστη κοινή υποσυμβολοσειρά (Longest Common Substring)

Έστω ότι δίνονται δύο συμβολοσειρές `X` και `Y` με μήκη `m` και `n` αντίστοιχα. Στο πρόβλημα της μέγιστης κοινής υποσυμβολοσειράς ζητείται να βρεθεί η μέγιστη σε μήκος υποσυμβολοσειρά που εντοπίζεται σε συνεχόμενες θέσεις των συμβολοσειρών `X` και `Y`. Για παράδειγμα αν `X="AATCGAG"` και `Y="CCATCGG"` τότε η μέγιστη κοινή υποσυμβολοσειρά είναι η `"ATCG"` με μήκος 4.

Γράψτε έναν απλοϊκό αλγόριθμο ωμής δύναμης (brute force) για την επίλυση του προβλήματος. Ο αλγόριθμος αυτός να δημιουργεί όλες τις υποσυμβολοσειρές του `X` 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 και τη ίδια τη μέγιστη υποσυμβολοσειρά που θα έχει εντοπιστεί.

2.2 Μέγιστη κοινή υποακολουθία (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 2. 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 μπορεί να βρεθεί στο https://www.cs.usfca.edu/ galles/visualization/DPLCS.html.

Algorithm 1 Αλγόριθμος μέγιστης κοινής υποσυμβολοσειράς

procedure LCSubstring(X, Y)
  $m \leftarrow length(X)$ ▷ m είναι το μήκος της συμβολοσειράς X
  $n \leftarrow length(Y)$ ▷ n είναι το μήκος της συμβολοσειράς Y
  for $i \leftarrow 1$ to m **do**
    $c[i, 0] \leftarrow 0$
  end for
  for $j \leftarrow 1$ to n **do**
    $c[0, j] \leftarrow 0$
  end for
  for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to n **do**
      if $X[i] == X[j]$ **then**
        $c[i, j] \leftarrow c[i - 1, j - 1] + 1$
      else
        $c[i, j] \leftarrow 0$
      end if
    end for
  end for
  Επέστρεψε το μήκος της μέγιστης κοινής υποσυμβολοσειράς ▷ Η μη μηδενική τιμή στο πλέον δεξί κάτω άκρο του πίνακα c είναι το μήκος της μέγιστης σε μήκος υποσυμβολοσειράς
end procedure

Algorithm 2 Αλγόριθμος μέγιστης κοινής υποακολουθίας (υπολογισμός μήκους)

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

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 α) σύμφωνα με τη μέγιστη κοινή υποσυμβολοσειρά και β) σύμφωνα με τη μέγιστη κοινή υποακολουθία. Εκτυπώστε τις ακολουθίες αυτές.

4 Παραδοτέα εργασίας

Τα παραδοτέα της εργασίας είναι τα ακόλουθα:

1. Κώδικας που υλοποιεί τους 4 αλγορίθμους που ζητούνται (ωμής δύναμης για τη μέγιστη κοινή υποσυμβολοσειρά, δυναμικού προγραμματισμού για τη μέγιστη κοινή υποσυμβολοσειρά, ωμής δύναμης για τη μέγιστη κοινή υποακολουθία, δυναμικού προγραμματισμού για τη μέγιστη κοινή υποακολουθία).
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 τους αλγόριθμους μέγιστης κοινής συμβολοσειράς και μέγιστης κοινής υποακολουθίας. Διερευνήστε αν υπάρχει η δυνατότητα περιορισμού των απαιτήσεων χώρου των αλγορίθμων δυναμικού προγραμματισμού για τα συγκεκριμένα προβλήματα.
   (γ′) Συνοπτική παρουσίαση των αποτελεσμάτων από την εκτέλεση του πειράματος.

5 Παρατηρήσεις

- Η υλοποίηση του κώδικα να γίνει, κατά προτίμηση, στη γλώσσα προγραμματισμού Python.
- Η εργασία είναι ατομική και η παράδοσή της γίνεται στο ecourse του μαθήματος μέχρι τις 13/02/2022.

Αναφορές

[Bha16] Aditya Bhargava. *Grokking Algorithms: An illustrated guide for programmers and other curious people*. Simon and Schuster, 2016.

[Bla20a] 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].

[Bla20b] Paul E. Black. Longest Common Substring. `https://xlinux.nist.gov/dads/HTML/longestCommonSubstring.html`, 2020. [Online; accessed 2-January-2022].