Πρώτο Σύνολο Ασκήσεων

2014-2015

Κατερίνα Ποντζόλκοβα, 5405

Αθανασία Ζαχαριά, 5295

Ερώτημα 1

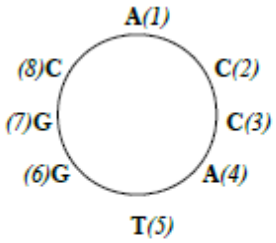
Μας δίνεται μια συλλογή από k ακολοθίες, k >=2 και αναζητούμε το πρότυπο P, μεγέθους n.

Ο αλγόριθμος εύρεσης επαναλήψεων ενός προτύπου σε κάθε ακολουθία είναι ο εξής:

1. Δημηούργησε ένα generalized suffix tree T, για την συλλογή από k ακολουθίες που μας έχει δοθεί.
2. Στη συνέχεια ξεκινώντας από την ρίζα, σύγκρινε έναν προς έναν τους χαρακτήρες του P, ακολουώντας το κατάλληλο μονοπάτι. Εάν εμφανιστεί κάποιο μη-ταίριασμα, τότε το πρότυπο δεν εμφανίζεται σε κάποια ακολουθία, διαφορετικά το πρότυπο εμφανίζεται και η λίστα των εμφανίσεων περιλαμβάνει όλα τα φύλλα του T, που βρισκεται κάτω από τον κόμβο του τελεύταιου χαρακτήρα του P.

Ερώτημα 2

Μας δίνεται μια γραμμική συμβολοσειρά α, μήκους n, και μια κυκλική συμβολοσειρά β



Εικόνα 1: Μια κυκλική συμβολοσειρά *β*.

Αναδιπλώνοντας την κυκλική συμβολοσειρά παίρνουμε την γραμμική συμβολοσειρά ACCATGGC.

Πριν εκτελέσουμε τον αλγόριθμο, πρέπει τα μήκη των 2 συμβολοσειρών να είναι συμβατά, δηλαδή η συμβολοσειρά β πρέπει να είναι ίση ή μεγαλύτερη από την α, με το μήκος της να είναι πολλαπλάσια του 8. Άρα διαιρούμε το n με το 8, και αν το υπόλοιπο είναι 0 τότε το μήκος της β συμβολοσειράς θα διαμορφοθεί: 8*πηλίκο(της διαίρεσης που κάναμε πριν). Αν το υπόλοιπο είναι διάφορο του μηδενώς τότε: 8*(πηλίκο+**1**). Άρα η αναδιπλούμενη συμβολοσειρά β που θα προκύψει είναι ACCATGGC... ACCATGGC.

Ύστερα μπορούμε να εφαρμόσουμε τον αλγόριθμο Knuth-Morris-Pratt ο οποίος έχει γραμμική πολυπλοκότητα. Ο αλγόριθμος αυτός συγκρίνει τους χαρακτήρες από αριστερά προς τα δεξιά χρησιμοποιώντας ένα ένα προ-επεξεργαστικό βήμα που στοιχίζει O(m) χώρο και χρόνο ένω η συνολική πολυπλοκότητα του αλγορίθμου είναι O(n+m) ανεξάρτητα από το μέγεθος του αλφαβήτου των ακολουθιών.

Ερώτημα 3

Έστω δύο συμβολοσειρές n χαρακτήρων και μια παράμετρος k. Σε κάθε συμβολοσειρά υπάρχουν n-k+1 υποσυμβολοσειρές μήκους k και Θ(n^2) ζευγάρια τέτοιων συμβολοσειρών, όπου κάθε στοιχείο του ζευγαριού ανήκει σε διαφορετική συμβολοσειρά. Θα χρησιμοποιήσουμε την μέθοδο του Δυναμικού Προγραμματισμού, η οποία αν και υλοποιείται εύκολα, με την χρήση πινάκων, παρουσιάζει ένα σημαντικό μειονέκτημα καθώς έχει μεγάλες απαιτήσιες στον χώρο αποθήκευσης. Θα δημιουργήσουμε έναν τετραγωνικό δισδιάστατο πίνακα n-k+1 επί n-k+1, το κάθε κελί του οποίου θα αντιπροσωπεύει ένα ζευγάρι από ακολουθίες. Ύστερα θα υπολογίσουμε την ομοιότητα των 2 ακολουθιών χρησιμοποιώντας την αναδρομική σχέση για τον υπλογισμό στοίχησης 2 ακολουθιών της υποενώτητας 4.3.2 των βοηθητικών σημειώσεων.

Ερώτημα 4

Έστω ένα γενικευμένο δέντρο επιθεμάτων όπου κάθε πλευρά έχει ετικέτα με έναν ή περισσότερους χαρακτήρες και ένα πρότυπο P. Αφού η ακολουθία δεν είναι γνωστή εκ των προτέρων, ενώ το πρότυπο είναι μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο αναζήτησης ακριβούς προτύπου, τον αλγόριθμο Knuth-Morris Pratt. Ο αλγόριθμος αυτός απαιτεί O(n) χρόνο προ-επεξεργασίας του προτύπου (m) χρόνο για την αναζήτηση. Λειτουργεί ως εξής: συγκρίνει τους χαρακτήρες από αριστερά προς τα δεξιά και κάθε φορά που βρίσκει ασυμφωνία στην σύγκριση, μετατοπίζει το πρότυπο δεξιά, με την υπόθεση ότι κάποιο πρόθεμα του προτύπου ταιριάζει με κάποιο επίθεμα κάποιου τμήματος της ακολουθίας.

Ερώτημα 5

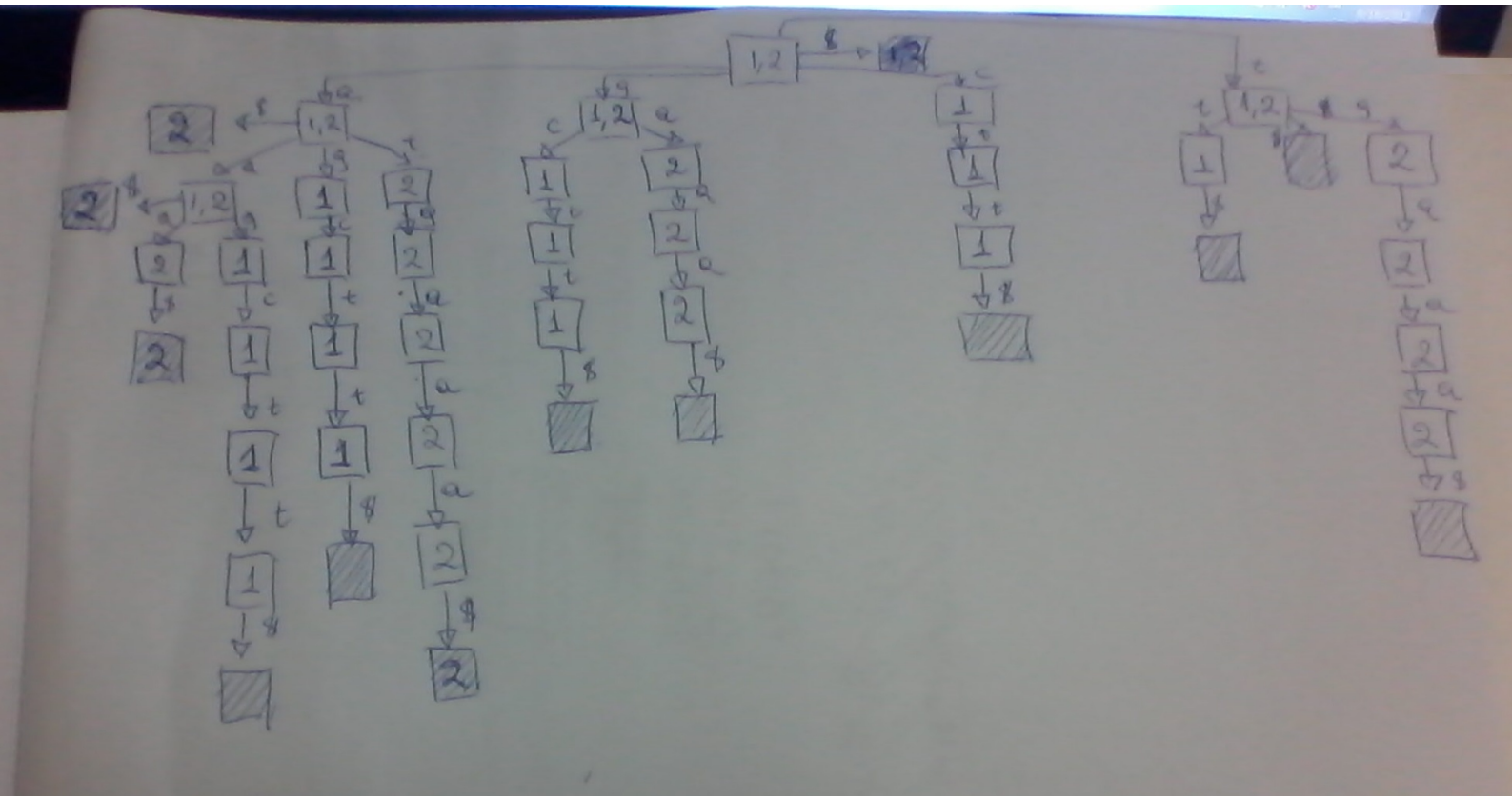
Έστω η ακολουθία με το ανθρώπινο γονιδίωμα S1 και με το γονιδίωμα του zebrafish S2.

Θα χρησιμοποιήσουμε δέντρο επιθεμάτων για τις ακολουθίες S1 και S2, αποθηκεύοντας όλα τα δυνατά επιθέματά τους. Έτσι κάθε φύλλο του δέντρου αναπαριστά είτε ένα επίθεμα μιας ακολουθίας είτε ένα κοινό επίθεμα που εμφανίζεται και στις 2 ακολουθίες. Σημειώνουμε κάθε εσωτερικό κόμβο του δέντρου u, με “1” ή με “2”, αν εμπεριέχει στο υποδέντρο του u κάποιο φύλλο που αναπαριστά κάποιο επίθεμα της ακολουθίας S1 ή S2. Η ετικέτα μονοπατιού -path label, κάθε εσωτερικού κόμβου που σημειώνεται ταυτόχρονα με “1” και “2”, αποτελεί κοινή υπό-συμβολοσειρά και η μεγαλύτερη σε μήκος είναι το ζητούμενο του προβλήματος. Οι πλήθος τέτοιων ετικετών είναι το πλήθος εμφανίσεων της μέγιστης υποσυμβολοσειράς.

Λόγω του μεγάλου μήκους των ζητούμενων ακολουθιών, θα πάρουμε ένα τμήμα τους 6 συμβόλων και θα κατασκευάσουμε ένα generalized suffix tree.

Έστω λοιπόν S1 = aagctt

S2 = atgaaa



Βλέπουμε ότι η μέγιστη υποσυμβολοσειρά είναι το aa και εμφανίζεται 1 φορά.

Ερώτημα 6

Δίνονται οι ακολουθίες v= ACCGATCGC και w= CCTGCACGTA.
Ολική στοίχιση:

D(i,j)		A	C	C	G	A	T	C	G	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
C	-1	\-1	←0	1	-1	-2	-3	-2	-3	-2
C	-2	-1	↑1	←2	1	0	-1	0	-1	0
T	-3	-2	0	↑1	1	0	1	0	-1	-1
G	-4	-3	-1	0	\2	1	0	0	1	0
C	-5	-4	0	1	1	\1	0	1	0	2
A	-6	-3	-1	0	0	↑2	←1	0	0	1
C	-7	-4	0	1	0	1	1	\2	1	2
G	-8	-5	-1	0	2	1	0	1	\3	2
T	-9	-6	0	-1	1	1	2	1	↑2	2
A	-10	-5	-1	-1	0	2	1	1	1	\1

A C C _ G _ A T C G _ C
_ C C T G C A _ C G T A

Τοπική στοίχηση:

D(i,j)		A	C	C	G	A	T	C	G	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
C	-1	-1	0	←1	-1	-2	-3	-2	-3	-2
C	-2	-1	1	↑2	1	0	-1	0	-1	0
T	-3	-2	0	\1	1	0	1	0	-1	-1
G	-4	-3	-1	0	\2	1	0	0	1	0
C	-5	-4	0	1	1	\1	0	1	0	2
A	-6	-3	-1	0	0	↑2	←1	0	0	1
C	-7	-4	0	1	0	1	1	\2	1	2
G	-8	-5	-1	0	2	1	0	1	\3	2
T	-9	-6	0	-1	1	1	2	1	2	2
A	-10	-5	-1	-1	0	2	1	1	1	1

C C _ G _ A T C G
C C T G C A _ C G

Ερώτημα 7

Έστω δυο συμβολοσειρές S_1 και S_2 και μια παράμετρος k. Θέλουμε να κατασκευάσουμε έναν αλγόριθμο γραμμικού χρόνου που να βρίσκει μια k-κάλυψη από δυο συμβολοσειρές ή να αποφαίνεται ότι τέτοια κάλυψη δεν υπάρχει.

Αρχικά θα υπολογίσουμε τα matching statistics ή ms(i) σε γραμμικό χρόνο χρησιμοποιώντας δέντρο επιθεμάτων. Κατασκευάζουμε ένα δέντρο επιθεμάτων T για την συμβολοσειρά S_2 χωρίς να αφαιρέσουμε τους συνδέσμους επιθεμάτων που χρησιμοποιούνται κατά την κατασκευή του δέντρου. Αυτό το δέντρο θα συντελέσει στην εύρεση των ms(i)

για κάθε θέση στη S_1 καθώς οι σύνδεσμοι επιθεμάτων χρησιμοποιούνται για την επιτάχυνση όλου του υπολογισμού.
Για να βρούμε το κάθε ms(i) συγκρίνουμε τους χαρακτήρες της S_1 με το δέντρο T ακολουθώντας το μονοπάτι του δέντρου T[1...m]. Το μήκος του μονοπατιού είναι το ms(1). Παίρνοντας την γενική περίπτωση ο αλγόριθμος ακολούθησε μια διαδρομή i<|m| για να υπολογίσει το ms(i). Αυτό σημαίνει ότι ο αλγόριθμος έχει εντοπίσει ένα σημείο b στο δέντρο T ώστε το μονοπάτι μέχρι εκείνο το σημείο να ταιριάζει με κάποιο επίθεμα του S_1 αλλά δεν επιτρέπεται περιταίρω ταίριασμα, ίσως επειδή ο αλγόριθμος έχει φτάσει σε κάποιο φύλλο.
Αφού υπολογίσουμε το ms(i), προχωράμε στον υπολογισμό του ms(i+1). Έχουμε τις ακόλουθες περιπτώσεις:

- αν το b είναι κόμβος που αντιστοιχεί στη ρίζα ο αλγόριθμος θα προχωρήσει από την ρίζα
- αν το b είναι εσωτερικός κόμβος τότε θα προχωρήσει από τον κόμβο b μέχρι τον κόμβο v
- αν το b δεν αντιστοιχεί σε εσωτερικό κόμβο, δηλαδή είναι φύλλο, τότε ο αλγόριθμος ανεβαίνει έναν κόμβο ακριβώς πάνω από το b.

Αφού υπολογίσει όλα τα ms(i), πρέπει να δούμε αν μπορούμε να συνθέσουμε το S2. Ξεκινάμε από την πρώτη θέση και ελέγχουμε αν το πρώτο γράμμα του εκάστοτε ms(i) ταιριάζει με το πρώτο γράμμα του S2. Αν δεν βρούμε κανένα ταίριασμα, ο αλγόριθμος τερματίζει με αποτυχία. Σε αντίθετη περίπτωση, με το πρώτο ms(i) που θα βρούμε, αποθηκεύουμε το μήκος του σε μια προσωρινή μεταβλητή m. Αρχίζουμε ξανά το ψάξιμο ψάχνοντας αυτή τη φορά τον χαρακτήρα S2(m+1). Αν τον βρούμε συνεχίζουμε την διαδικασία μέχρι να φτάσουμε στο τέλος του S2, όπου το πρόγραμμα θα τερματίσει με επιτυχία. Αν δεν τον βρούμε μειώνουμε το m κατά 1, και μειώνουμε και το ms(i) κατά ένα για να μην έχουμε επικάλυψη στη συμβολοσειρά S2. Αν το m φτάσει στο 0, το πρόγραμμα τερματίζει με αποτυχία