

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΤΕΚΜΗΡΙΩΣΗ ΑΠΑΛΛΑΚΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

ΜΑΘΗΜΑ

ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

ΕΠΙΜΕΛΕΙΑ ΕΡΓΑΣΙΑΣ

ΦΑΤΙΟΝ ΚΡΙΣΤΑΚΙ, Π20107

ΑΛΕΞΑΝΔΡΟΣ ΧΟΛΗΣ, Π20217

ΙΩΑΝΝΗΣ ΒΛΑΜΗΣ, Π20033

ΠΕΙΡΑΙΑΣ 2023

Περιεχόμενα

Γενικές οδηγίες 3

Θέμα 1 4

Θέμα 2 5

Θέμα 3 6

Βιβλιογραφία 7

Γενικές οδηγίες

Στο αρχείο source2023.zip θα βρείτε ένα φάκελο (projects) στον οποίο εμπεριέχονται τρεις υποφάκελοι για κάθε θέμα αντίστοιχα (sub\_ex\_1, sub\_ex\_2, sub\_ex\_3) καθώς και ένας υποφάκελος με τα βιολογικά δεδομένα. Επίσης υπάρχει και ένα txt αρχείο (requirements.txt) όπου βρίσκονται όλες οι βιβλιοθληκες της python που έχουν χρησιμοποιηθεί. Προκειμένου να εκτελέσετε κάθε πρόγραμμα ξεχωριστά, αρκεί να τρέξετε το αρχείο main.py σε κάθε υποφάκελο.

Θέμα 1

Για αυτό το θέμα χρησιμοποιήθηκε η θεωρία για το Hidden Markov Model (HMM) όπου με χρήση δυναμικού προγραμματισμού κατασκευάζουμε το scoring matrix, το οποίο περιέχει 4 στήλες (όσο το μήκος της αλληλουχίας GGCT) και 2 γραμμές (όσες οι καταστάσεις που έχουμε). Η πρώτη στήλη του πίνακα συμπληρώνεται με βάση τον τύπο log2(x) + log2(y), όπου x η πιθανότητα η συγκεκριμένη κατάσταση να είναι αρχική και y η πιθανότητα η συγκεκριμένη κατάσταση να εκπέμψει το σύμβολο που παρουσιάζεται. Κάνουμε την παραδοχή ότι είναι ισοπίθανα οι καταστάσεις να είναι αρχικές. Από την δεύτερη στήλη και μετά ακολουθούμε το εξής μοτίβο. Σε κάθε θέση του πίνακα βάζουμε την μέγιστη τιμή της προηγούμενης στήλης + τον λογάριθμο πιθανότητας μετάβασης κατάστασης + τον λογάριθμο η νέα κατάσταση να εκπέμψει το συγκεκριμένο σύμβολο.

Αφού συμπληρωθεί ο πίνακας, κάνουμε backtraking με βάση τη μέγιστη τιμή κάθε στήλης και αποθηκεύουμε την κατάσταση.

Στον υποφάκελο sub\_ex\_1 όπου βρίσκεται το πρώτο θέμα θα βρείτε 3 αρχεία python. Στο αρχείο probabilities.py περιέχονται όλες οι πιθανότητες που θα χρειαστούμε για τον υπολογισμό του scoring matrix. Στο αρχείο viterbi.py περιέχονται δυο συναρτήσεις. Η πρώτη συνάρτηση (get\_scores /1) δέχεται σαν όρισμα την αλληλουχία που μας δίνεται και κατασκευάζει και επιστρέφει το scoring martix. Η δεύτερη συνάρτηση (get\_path /2) δέχεται σαν ορίσματα το scoring matrix και την αλληλουχία και επιστρέφει τις πιθανές καταστάσεις για τη συγκεκριμένη αλληλουχία.

Θέμα 2

Για αυτό το θέμα χρησιμοποιήσαμε την βασική ιδέα από το παράδειγμα με το παιχνίδι με τις “Πέτρες” στην εισαγωγή του βιβλίου “Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής”. Δηλαδή κατασκεύαζουμε έναν πίνακα NxM (όπου N και M τα μήκοι των δύο αλληλουχιών) με βάση αυτού οι δύο παίκτες κάνουν τις κινήσεις τους. Κάνουμε την παραδοχή ότι σε κάθε γύρο ο υπολογιστής ‘βγάζει’ έναν τυχαίο αριθμό όπου ο παίκτης πρέπει να αφαιρέσει. Σκοπός του κάθε παίχτη είναι να στήλει τον αντίπαλο σε κατάσταση Lose ή αν μπόρει να τελειώσει το παιχνίδι (κατάσταση Finish). Ο πίνακας συμπληρώνεται ως εξής: η θέση 0,0 υποδηλώνει το τέλος του παιχνιδιού. Έπειτα η πρώτη γραμμή και στήλη συμπλήρώνεται με κατάσταση Win διότι ο παίκτης έχει την δυνατότητα να νικήσει. Ο υπόλοιπος πίνακας συμπληρώνεται σύμφωνα με το αν το άθροισμα της στήλης και της γραμμής είναι ζυγός αριθμός, τότε είναι κατάσταση Win, αλλιώς είναι κατάσταση Lose.

Στον υποφάκελο sub\_ex\_2 όπου βρίσκεται το δεύτερο θέμα θα βρείτε 3 αρχεία python. Στο αρχείο algorithm.py περιέχονται δυο συναρτήσεις. Η πρώτη συνάρτηση (create\_matrix /2) δέχεται σαν ορίσματα τις δύο αλληλουχίες, δημιουργεί και επιστρέφει τον πίνακα που περιγράψαμε παραπάνω. Η δεύτερη συνάρτηση (play /1) δέχεται σαν όρισμα τον πίνακα που δημιουργήσαμε. Αρχικά ‘βγάζει’ έναν τυχαίο αριθμό (ο οποίος είναι το πόσα στοιχεία πρέπει να αφαιρέσει ο παίκτης). Έπειτα κοιτάμε τα σύμβολα που υπάρχουν Χ θέσεις προς τα πάνω (αφαιρούμε από την πρώτη αλληλουχία), προς τα αριστερά (αφαιρούμε από την δεύτερη αλληλουχία) και διαγώνια (αφαιρούμε και από τις δύο αλληλουχίες), και ελέγχουμε αν υπάρχει κάπου κατάσταση Lose την οποία επιλέγουμε. Αν δεν υπάρχει κάποια τέτοια κατάσταση τότε επιλέγουμε κατάσταση Win. Στο τέλος εμφανίζουμε ποιός παίκτης νίκησε, δηλαδή ποιος θα φτάση πρώτος στην πρώτη θέση του πίνακα.

Θέμα 3

Για την λύση του θέματος 3 χρησιμοποιήσαμε την βασική ιδέα από το παράδειγμα με το παιχνίδι με τις “Πέτρες” στην εισαγωγή του βιβλίου “Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής”. Με την χρήση δυναμικού προγραμματισμού, κατασκεύαζουμε έναν πίνακα NxM (όπου N και M τα μήκοι των δύο αλληλουχιών) και με βάση τον πίνακα αυτό, οι δύο παίκτες κάνουν τις κινήσεις τους.

Η βασική ιδέα δημιουργίας του πίνακα είναι η χρήση τριών συμβόλων (‘F’ για Finish, ‘W’ για Win, ‘L’ για Lose). Η πρώτη στήλη και γραμμή αρχικοποιούνται με το γράμμα ‘F’ καθώς σε περίπτωση που ένας παίκτης οδηγηθεί σε αυτή τη κατάσταση δεν μπορεί να κάνει κάποια κίνηση και αυτομάτως χάνει. Οι στήλες 2, 3 καθώς και οι γραμμές 2, 3 (με εξαίρεση τις θέσεις του πίνακα που έχουν αρχικοποιηθεί ως ‘F’) αρχικοποιούνται ως ‘W’ αφού ο παίκτης σε αυτή τη κατάσταση μπορεί να στείλει τον αντίπαλο σε κατάσταση ‘F’. Όλες οι υπόλοιπες θέσεις σημπληρώνονται ως εξής: Εαν από την συγκεκριμένη κατάσταση οδηγούμαστε αποκλειστικά σε κατάσταση ‘W’ τότε συμπληρώνεται ως ‘L’ αλλιώς ‘W’.

Αφού δημιουργηθεί ο πίνακας αρχίζει ο κάθε παίκτης να αφαιρεί σύμβολα. Ξεκινάμε από την τελευταία θέση του πίνακα και προσπαθούμε να στείλουμε το παιχνίδι σε κατάσταση ‘F’ όπου ο αντίπαλος δεν θα μπορεί να κάνει κάποια κίνηση και θα κερδίσουμε. Σε περίπτωση που δεν υφίσταται ακόμα ο τερματισμός, προσπαθούμε να στείλουμε τον αντίπαλο σε κατάσταση ‘L’ εαν αυτό είναι δυνατό ώστε να χάσει. Λόγω της μη τυχαιότητας των κινήσεων (όπως στο θέμα 2) γνωρίζουμε εξ αρχής ποιος παίκτης θα νικήσει. Εαν η κατάσταση του παίκτη που ξεκινάει είναι ‘W’ τότε ο παίκτης που ξεκινάει κερδίζει πάντα.

Στον υποφάκελο sub\_ex\_3 όπου βρίσκεται το τρίτο θέμα θα βρείτε 3 αρχεία python. Το αρχείο main.py (το οποίο εκτελείτε), το αρχείο read\_data.py (το οποίο διαβάζει τα δεδομένα μας) και το algorithm.py όπου βρίσκεται όλη η λογική του αλγορίθμου μας. Εκεί θα βρείτε δύο συναρτήσεις. Την create\_matrix /2 και την play /1. Η create\_matrix δέχεται τις δύο ακολουθίες και δημιουργεί και επιστρέφει τον πίνακα που αναφέραμε πιο πάνω. Η play δέχεται τον πίνακα αυτόν και βρίσκει την επόμενη κίνηση που θα γίνει βάση της κατάστασης που βρίσκεται. Ελέγχουμε πρώτα εαν μπορούμε να κάνουμε κίνηση και σε περίπτωση που μπορούμε βρίσκουμε κατάσταση ‘F’ ή ‘L’. Σε περίπτωση που δεν υπάρχει αναγκαστικά στέλνουμε τον αντίπαλο σε κατάσταση νίκης. Τέλος εκτυπώνουμε τον νικητή του παιχνιδιού.

Βιβλιογραφία

* ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ ΒΙΟΠΛΗΡΟΦΟΡΙΚΗΣ, NEIL C. JONES, PAVEL A. PEVZNER
* Βιοπληροφορική & Λειτουργική Γονιδιωματική, Jonathan Pevsner