

**Π16036** – Ιωαννίδης Παναγιώτης

**Π16112** – Παραβάντης Αθανάσιος

|  |
| --- |
| Εργασία |
| **Βιοπληροφορική** |
| Ακαδημαϊκό έτος 2018 - 2019 |

Περιεχόμενα

[1 Άσκηση 6.14 3](#_Toc12720359)

[1.1 Εκφώνηση 3](#_Toc12720360)

[1.2 Σχεδιασμός 3](#_Toc12720361)

[1.3 Υλοποίηση 3](#_Toc12720362)

[1.4 Αποτέλεσμα 4](#_Toc12720363)

[1.5 Αναφορές 4](#_Toc12720364)

[2 Άσκηση 6.15 5](#_Toc12720365)

[2.1 Εκφώνηση 5](#_Toc12720366)

[2.2 Σχεδιασμός 5](#_Toc12720367)

[2.3 Υλοποίηση 5](#_Toc12720368)

[2.4 Αποτέλεσμα 5](#_Toc12720369)

[2.5 Αναφορές 6](#_Toc12720370)

[3 Άσκηση 6.22 7](#_Toc12720371)

[3.1 Εκφώνηση 7](#_Toc12720372)

[3.2 Σχεδιασμός 7](#_Toc12720373)

[3.3 Υλοποίηση 7](#_Toc12720374)

[3.4 Αποτέλεσμα 8](#_Toc12720375)

[3.5 Αναφορές 8](#_Toc12720376)

[4 Άσκηση 6.27 9](#_Toc12720377)

[4.1 Εκφώνηση 9](#_Toc12720378)

[4.2 Σχεδιασμός 9](#_Toc12720379)

[4.3 Υλοποίηση 9](#_Toc12720380)

[4.4 Αποτέλεσμα 9](#_Toc12720381)

[4.5 Αναφορές 10](#_Toc12720382)

[5 Άσκηση 6.37 11](#_Toc12720383)

[5.1 Εκφώνηση 11](#_Toc12720384)

[5.2 Σχεδιασμός 11](#_Toc12720385)

[5.3 Υλοποίηση 11](#_Toc12720386)

[5.4 Αποτέλεσμα 12](#_Toc12720387)

[5.5 Αναφορές 12](#_Toc12720388)

[6 Άσκηση 11.4 13](#_Toc12720389)

[6.1 Εκφώνηση 13](#_Toc12720390)

[6.2 Σχεδιασμός 13](#_Toc12720391)

[6.3 Υλοποίηση 13](#_Toc12720392)

[6.4 Αποτέλεσμα 15](#_Toc12720393)

[6.5 Αναφορές 15](#_Toc12720394)

[7 Άσκηση 11.6 16](#_Toc12720395)

[7.1 Εκφώνηση 16](#_Toc12720396)

[7.2 Σχεδιασμός 16](#_Toc12720397)

[7.3 Υλοποίηση 17](#_Toc12720398)

[7.4 Αποτέλεσμα 18](#_Toc12720399)

[7.5 Αναφορές 19](#_Toc12720400)

[8 Βοηθητικά αρχεία 20](#_Toc12720401)

# Άσκηση 6.14

## Εκφώνηση

Δύο παίκτες παίζουν το εξής παιχνίδι με δύο αλληλουχίες που έχουν μήκος n και m νoυκλεοτίδια αντίστοιχα. Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει δύο νoυκλεοτίδια από τη μία αλληλουχία (είτε την πρώτη είτε τη δεύτερη) και ένα νουκλεοτίδιο από την άλλη. Ο παίκτης που δεν μπορεί να κάνει κίνηση κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές των n και m.

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7.

## Υλοποίηση

Η άσκηση 6.14 έχει βασιστεί στην παραδοχή ότι δυο παίχτες παίζουν μεταξύ τους. Έτσι, λοιπόν, έχει γίνει προσομοίωση των παιχτών, τους οποίους χειρίζεται ο υπολογιστής. Ο πρώτος παίχτης παίζει με στρατηγική που σκοπό έχει να τον οδηγήσει σε κατάσταση νίκης. Από την άλλη οι κινήσεις του δεύτερο παίχτη είναι εντελώς τυχαίες.

Έστω, λοιπόν, m η ακολουθία 1 και n η ακολουθία 2. Ο πρώτος παίκτης θα βρίσκεται σε μειονεκτική θέση όταν:

m = 3k and n ≥ 3k για k ∈ N

n = 3k and m ≥ 3k για k ∈ N

m = n = 3k + 1 για k ∈ Ν

Για παράδειγμα:

m = 0 and n ≥ 0, m = 3 και n ≥ 3, m = 6 και n ≥ 6, . . .

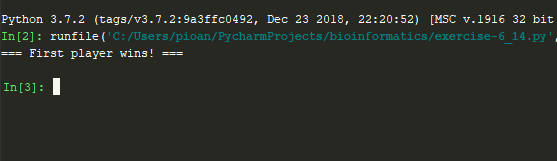
n = 0 και m ≥ 0, n = 3 και m ≥ 3, n = 6 και m ≥ 6, . . .

m = n = 1, m = n = 4, m = n = 7, . . .

Επομένως εμείς με τον με πρέπει να τον φέρουμε σε πλεονεκτικές, για εκείνον, θέσεις.  
Η ακολουθίες από επιλέξαμε για την εκτέλεση της άσκησης είναι αυτές της α-λακταλβουμίνη και της λυσοζύμης c.

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6\_14.py**.   
Το αποτέλεσμα που εμφανίζεται είναι ο παίκτης που νικάει το παιχνίδι, όπως φαίνεται και στην εικόνα 1.1.



Εικόνα 1.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

# Άσκηση 6.15

## Εκφώνηση

Δύο παίκτες παίζουν το παρακάτω παιχνίδι με μια νουκλεοτιδική αλληλουχία που έχει μήκος n. Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει είτε ένα είτε δύο νoυκλεοτίδια από την αλληλουχία, Ο παίκτης που αφαιρεί το τελευταίο γράμμα κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές του n.

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7.

## Υλοποίηση

Και αυτό το παιχνίδι αποτελείται από δυο παίχτες, οι οποίοι προσομοιώνονται από τον υπολογιστή. Η ακολουθία που επιλέξαμε για το παιχνίδι είναι η λυσοζύμης c.

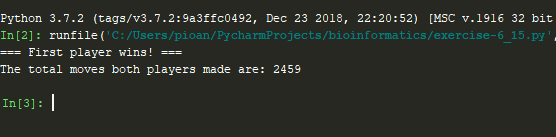
Ο πρώτος παίκτης επιλεγεί την κίνηση του με βάση το υπόλοιπο της διαίρεσης του πλήθους των νουκλεοτίδιο, που υπάρχουν κάθε στιγμή στη ακολουθία, και του 3. Αν το υπόλοιπο είναι ίσο με μηδέν τότε επιλεγεί να αφαίρεση ένα νουκλεοτίδιο, αν το υπόλοιπο είναι ίσο με ένα τότε επιλεγεί να αφαιρέσει δυο νoυκλεοτίδια αλλιώς η επιλογή γίνεται στην τύχη.

Ο επιλογές του δευτέρου παίκτη είναι καθαρά τυχαίες.

Με αυτόν τον τρόπο, θέτουμε τον πρώτο παίκτη σε πλεονεκτική θέση η οποία θα τον οδηγήσει στη νίκη.

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6\_15.py**.   
Το αποτέλεσμα που εμφανίζεται είναι ο παίκτης που νικάει το παιχνίδι καθώς και οι συνολικές κίνησης και των δυο παικτών, εικόνα 2.1.



Εικόνα 2.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* <https://en.wikipedia.org/wiki/Nim>

# Άσκηση 6.22

## Εκφώνηση

Ορίζουμε ότι η στοίχιση επικάλυψης μεταξύ δυο αλληλουχιών **v** = v1…vn και **w** = w1…wm είναι η στοίχιση ανάμεσα σε ένα πρόθεμα της **v** και ένα επίθεμα της **w**. Για παράδειγμα, αν **v** = TATATA και **w** = AAATTT, τότε μια (όχι απαραιτήτως βέλτιστη) στοίχιση επικάλυψης μεταξύ των **v** και **w** είναι η

ATA

AAA

Η βέλτιστη στοίχιση επικάλυψης είναι η στοίχιση που μεγιστοποιεί τη βαθμολογία της καθολικής στοίχισης μεταξύ των vi,…,vn και w1,…,wj, όπου το μέγιστο υπολογίζεται για όλα τα προθέματα vi,…,vn της **v** και όλα τα επιθέματα w1,…,wj της **w**.

Διατυπώστε έναν αλγόριθμο που υπολογίζει τη βέλτιστη στοίχιση επικάλυψης και εκτελείται σε χρόνο O(nm).

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

## Υλοποίηση

Για να υπολογίσουμε τη βέλτιστη στοίχιση επικάλυψης δυο ακολουθιών σύμφωνα με τα ζητούμενα της άσκησης, το πρώτο βήμα που πρέπει να κάνουμε είναι να βρούμε όλα τα πιθανά προθέματα της v = “TATATA” και όλα τα πιθανά επιθέματα της w = “AAATTT”. Αυτό μπορούμε να το επιτύχουμε εύκολα, δημιουργώντας δυο λίστες οι οποίες περιέχουν τους κατάλληλους συνδυασμούς υποακολουθιών προς σύγκριση. Οι υποακολουθίες αυτές, έχουν μήκος από n = 1 μέχρι και n = N – 1 όπου το N είναι το μήκος της αρχικής ακολουθίας v ή w.

Έπειτα, αφού έχουμε δεδομένες όλες τις πιθανές υποακολουθίες, μπορούμε να αρχίσουμε τη διαδικασία στοίχισης βασιζόμενοι στη καθολική στοίχιση των Needleman-Wunsch. Για να έχουμε όμως τα επιθυμητά αποτελέσματα, οφείλουμε να κάνουμε μια τροποποίηση στη βαθμολόγηση του αλγορίθμου. Όταν υπάρχει αντιστοιχία μεταξύ δυο νουκλεοτιδίων, επιβραβεύουμε τον αλγόριθμο με +1 και σε κάθε άλλη περίπτωση τιμωρούμε τον αλγόριθμο με -1. Επομένως, θα έχουμε έναν σταθερό τρόπο βαθμολόγησης που θα μας βοηθήσει να βρούμε την βέλτιστη στοίχιση επικάλυψης.

Τέλος, συγκρίνουμε μία προς μία όλες τις υποακολουθίες που μας αφορούν με βάση το μοντέλο που περιγράψαμε και συμπεραίνουμε με βάση τα κριτήρια βαθμολόγησης, ποιες υποακολουθίες έχουν την καλύτερη επίδοση. Αφού βρούμε μια v’ και μια w’ εμφανίζουμε τη καθολική στοίχιση στην οθόνη ως αποτέλεσμα.

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6\_22.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνα 3.1

A screenshot of a computer

Description automatically generated

Εικόνα 3.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

# Άσκηση 6.27

## Εκφώνηση

Για μια παράμετρο k, υπολογίστε την καθολική στοίχιση δύο συμβολοσειρών, με τον περιορισμό ότι η στοίχιση περιέχει το πολύ k κενά (μπλοκ με συνεχόμενες προσθαφαιρέσεις).

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

## Υλοποίηση

Ξεκινάμε την επίλυση της άσκησης με δυο τυχαίες ακολουθίες v = “ATCTGATAA” και w = “TGCATA” τις οποίες θα τις συγκρίνουμε με καθολική στοίχιση βασιζόμενοι σε μια παράμετρο k που μας δίνεται από τον χρήστη. Για να δούμε εάν μια από τις δυο ακολουθίες περιέχουν προσθαφαιρέσεις, τροποποιούμε τις επιβραβεύσεις και τις ποινές τους αλγορίθμου Needleman-Wunsch ώστε να δίνουμε +0 για ταίριασμα ή ασυμφωνία και -1 σε κάθε άλλη περίπτωση.

Έχοντας υπ’ όψη αυτό το τρόπο σύγκρισης και τη παράμετρο k μπορούμε να διαμορφώσουμε ένα κριτήριο ανάλογα με το πόσα κενά ή προσθαφαιρέσεις επιθυμούμε. Εάν για παράδειγμα το k = 2 και η βαθμολογία σύγκρισης είναι -3, αυτό σημαίνει ότι μια από τις δυο ακολουθίες περιέχουν 3 κενά ή προσθαφαιρέσεις και πρέπει να σταματήσουμε εδώ. Αντιθέτως, εάν k = 3 ή παραπάνω τότε αυτό υποδεικνύει στο πρόγραμμα ότι είμαστε εντάξει με τη βαθμολογία που προκύπτει.

Τελικά, αν το k είναι μεγαλύτερο από τη βαθμολογία (από αρνητική τη κάνουμε θετική πάντα) τότε προχωράει το πρόγραμμα στην εκτύπωση όλων των πιθανών στοιχίσεων των δυο ακολουθιών. Σε κάθε άλλη περίπτωση, εμφανίζεται μήνυμα ότι ο έλεγχος απέτυχε και έτσι τερματίζει το πρόγραμμα.

## Αποτέλεσμα

Στην παρακάτω εικόνα έχουμε δυο στιγμιότυπα εκτέλεσης. Στη πρώτη περίπτωση, το k = 2 και η βαθμολογία = 3 επομένως σταματάει και μένει το πρόγραμμα ως έχει χωρίς στοίχιση. Στη δεύτερη περίπτωση, έχουμε k = 3 και βαθμολογία = 3 άρα συνεχίζουμε και στοιχίζουμε τις ακολουθίες v και w καθολικά.

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6\_27.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνα 4.1.

A screenshot of a cell phone

Description automatically generated

Εικόνα 4.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* <https://en.wikipedia.org/wiki/Longest_common_subsequence_problem>

# Άσκηση 6.37

## Εκφώνηση

Στο πρόβλημα της Χιμαιρικής Στοίχισης, δίνονται μια συμβολοσειρά **v** και ένα σύνολο συμβολοσειρών {w1,…,wN}, και πρέπει να βρεθεί το max1≤i,j≤N s(v,wi o wj), όπου wi o wj είναι η συνένωση των **w**i και **w**j και το s(.,.) συμβολίζει τη βαθμολογία της βέλτιστης καθολικής στοίχισης.

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

## Υλοποίηση

Στην επίλυση του προβλήματος της Χιμαιρικής Στοίχισης χρησιμοποιούμε την ακολουθία v = “AGTACT” και το σύνολο συμβολοσειρών w = { “ATAGAC”, “CTGATA”, “ACGGGT” } ως δεδομένα για την άσκηση. Για να βρούμε τη βέλτιστη καθολική στοίχιση μεταξύ της ακολουθίας v και μια συνένωση δυο ακολουθιών του w, δημιουργούμε λίστα με όλους τους συνδυασμούς των συμβολοσειρών της w.

Χρησιμοποιούμε τη λίστα των συνδυασμών της w για να εφαρμόσουμε καθολικές στοιχίσεις με τη v. Με αυτό τον τρόπο, συλλέγουμε όλες τις βαθμολογίες του αλγορίθμου Needleman-Wunsch και επιλέγουμε το ζευγάρι με τη μέγιστη απόδοση. Η βαθμολογία ορίζεται ως +1 για ταίριασμα και -1 σε κάθε άλλη περίπτωση.

Τέλος, αφού επιλέξουμε την ακολουθία v και την συνένωση δυο ακολουθιών του w, έχοντας εξασφαλίσει τη μέγιστη βαθμολογία, εκτυπώνουμε όλες τις δυνατές στοιχίσεις των δυο ακολουθιών.

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6\_37.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνα 5.1.

A screenshot of a cell phone

Description automatically generated

Εικόνα 5.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* <https://en.wikipedia.org/wiki/Longest_common_subsequence_problem>

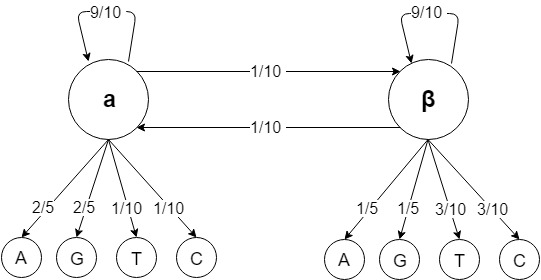
# Άσκηση 11.4

## Εκφώνηση

Στο σχήμα 1 φαίνεται ένα ΗΜΜ με δύο καταστάσεις α και β. Όταν το ΗΜΜ βρίσκεται στην κατάσταση α, έχει μεγαλύτερη πιθανότητα να εκπέμψει πουρίνες

(Α και G) . Όταν βρίσκεται στην κατάσταση β έχει μεγαλύτερη πιθανότητα να εκπέμψει πυριμιδίνες (C και T ). Αποκωδικοποιήστε την πιο πιθανή ακολουθία

των καταστάσεων (α/β) για την αλληλουχία **GGCT** Χρησιμοποιήστε λογαριθμικές βαθμολογίες αντί για κανονικές βαθμολογίες πιθανοτήτων .



Σχήμα 1

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και η βιβλιοθήκη numpy.

## Υλοποίηση

Για να αποκωδικοποιήσουμε την ακολουθία των καταστάσεων (α/β) για την αλληλουχία **GGCT** θα χρησιμοποιήσουμε τον αλγόριθμο του Viterbi και θα στηριχτούμε στις αρχές του δυναμικού προγραμματισμού.

Αρχικά αποκωδικοποιούμε τα δεδομένα που μας παρέχει το πρόβλημα και σχηματίζουμε τους απαραίτητους πίνακες, εικόνα 4.1.

|  |  |  |
| --- | --- | --- |
| από/προς | α | β |
| START | 0.50 | 0.50 |
| α | 0.90 | 0.10 |
| β | 0.10 | 0.90 |

Πίνακας 1 – Πίνακας μεταβάσεων

|  |  |  |
| --- | --- | --- |
| εκπομπή/κατάσταση | α | β |
| Α | 0.40 | 0.20 |
| G | 0.40 | 0.20 |
| T | 0.10 | 0.30 |
| C | 0.10 | 0.30 |

Πίνακας 2 – Πίνακας εκπομπών

Όπως φαίνεται και στο σχήμα 1 το ΗΜΜ αποτελείται από δυο καταστάσεις, την κατάσταση α και την κατάσταση β. Υπάρχουν πάρα πολλά μονοπάτια που οδηγούν στην ζητούμενη κατάσταση αλλά εμείς θα υπολογίζουμε αυτό με την μεγαλύτερη πιθανότητα.

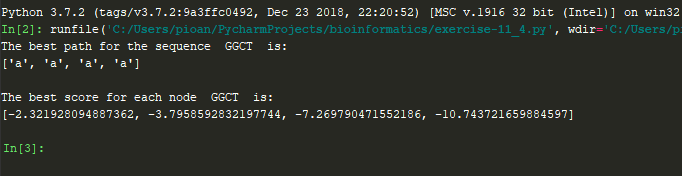
Για τους υπολογισμούς μας, είναι αποδοτικότερο να χρησιμοποιήσουμε τον λογάριθμο των πιθανοτήτων. Έτσι, θα πρέπει να υπολογίσουμε το άθροισμα των πιθανοτήτων αντί για το γινόμενο τους.

Θεωρούμε ότι η πιθανότητα για την επιλογή αρχικής κατάστασης είναι 1/2 για την μετάβαση στην κατάσταση α και 1/2 για την μετάβαση στην κατάσταση β.

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-11\_4.py**.

Με βάση τα δεδομένα μας, η βέλτιστη αλληλουχία καταστάσεων για την αλληλουχία **GGCT** είναι η αααα.   
Στην εικόνα 6.1 παρουσιάζεται το αποτέλεσμα της εκτέλεσης του προγράμματος. Εκτός από την βέλτιστη αλληλουχία καταστάσεων παρουσιάζονται και οι μέγιστες πιθανότητες για κάθε κόμβο της διαδρομής καταστάσεων που επιλέχθηκε.



Εικόνα 6.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Αναγνώριση προτύπων

Σ. Θεοδωρίδης, Κ. Κουτρούμπας

* Σημειώσεις Αναγνώριση Προτύπων, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* <https://en.wikipedia.org/wiki/Viterbi_algorithm>

# Άσκηση 11.6

## Εκφώνηση

Θεωρήστε ένα διαφορετικό παιχνίδι στο οποίο κρουπιέρης δεν ρίχνει νόμισμα αλλά ζάρι με τρεις πλευρές που έχουν ετικέτες 1,2 , και 3. (Μην προσπαθήσετε να σκεφτείτε την εμφάνιση ενός τέτοιου ζαριού.) Ο κρουπιέρης έχει δύο στημένα

ζάρια D1 και D2. Για κάθε ζάρι D, η πιθανότητα να προκύψει ο αριθμός είναι ίση

με 1/2, και η πιθανότητα για τα άλλα δύο αποτελέσματα είναι ίση με 1/4. Σε κάθε γύρο , ο κρουπιέρης πρέπει να αποφασίσει αν (1) θα κρατήσει το ίδιο ζάρι, (2) θα αλλάξει ζάρι , ή (3) θα σταματήσει το παιχνίδι . Επιλέγει το (1) με πιθανότητα 1/2 και τα (2) και (3) με πιθανότητα 1/4. Στην αρχή, ο κρουπιέρης επιλέγει ένα από τα δύο ζάρια με την ίδια πιθανότητα.

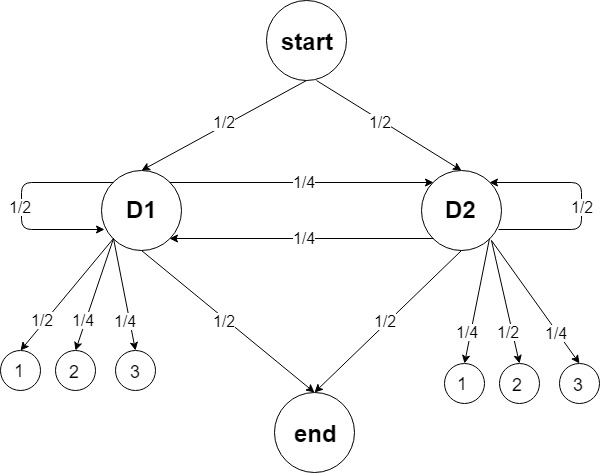
* Διατυπώστε ένα ΗΜΜ για την παραπάνω κατάσταση. Προσδιορίστε το αλφάβητο, τις καταστάσεις, τις πιθανότητες μεταβολής κατάστασης, και τις πιθανότητες εκπομπής. Συμπεριλάβετε μια αρχική κατάσταση start, και υποθέστε ότι το ΗΜΜ ξεκινάει στην κατάσταση start με πιθανότητα 1. Συμπεριλάβετε επίσης και μια τελική κατάσταση end .
* Ας υποθέσουμε ότι παρατηρείτε την εξής ακολουθία από ρίψεις ζαριών : **112122**. Βρείτε μια ακολουθία καταστάσεων που εξηγεί καλύτερα την ακολουθία των ρίψεων. Ποια είναι η πιθανότητα της συγκεκριμένης ακολουθίας; Βρείτε την απάντηση συμπληρώνοντας τον πίνακα Viterbi. Συμπεριλάβετε βέλη οπισθοδρόμησης στα κελιά έτσι ώστε να είστε σε θέση να ανιχνεύσετε αντίστροφα την ακολουθία των καταστάσεων . Μερικά από τα παρακάτω δεδομένα ίσως φανούν χρήσιμα :
* Υπάρχουν στην πραγματικότητα δύο βέλτιστες ακολουθίες καταστάσεων για τη συγκεκριμένη ακολουθία ρίψεων των ζαριών. Ποια είναι η άλλη ακολουθία καταστάσεων;

## Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και η βιβλιοθήκη numpy.

## Υλοποίηση

Στο σχήμα 2 παρουσιάζεται το ΗΜΜ που περιγράφεται στη εκφώνηση της άσκησης και στους πίνακες 3 και 4 τα αντίστοιχα δεδομένα.



Σχήμα 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| από/προς | START | D1 | D2 | END |
| START | 0.00 | 0.50 | 0.50 | 0.00 |
| D1 | 0.00 | 0.50 | 0.25 | 0.25 |
| D2 | 0.00 | 0.25 | 0.50 | 0.25 |
| END | 0.00 | 0.00 | 0.00 | 0.00 |

Πίνακας 3 – Πίνακας μεταβάσεων

|  |  |  |  |
| --- | --- | --- | --- |
| κατάσταση/εκπομπή | 1 | 2 | 3 |
| START | 0.00 | 0.00 | 0.00 |
| D1 | 0.5 | 0.25 | 0.25 |
| D2 | 0.25 | 0.50 | 0.25 |
| END | 0.00 | 0.00 | 0.00 |

Πίνακας 4 – Πίνακας εκπομπών

Και αυτή η άσκηση θα υλοποιηθεί με τη μέθοδο του δυναμικού προγραμματισμού και του αλγορίθμου Viterbi.

Για τους υπολογισμούς μας, είναι αποδοτικότερο να χρησιμοποιήσουμε τον λογάριθμο των πιθανοτήτων. Έτσι, θα πρέπει να υπολογίσουμε το άθροισμα των πιθανοτήτων αντί για το γινόμενο τους.

Δεδομένου ότι δεν αρκεί να βρούμε μόνο την βέλτιστη ακολουθία θα πρέπει να υπολογίσουμε τις πιθανότητες για όλες τις αλλαγές καταστάσεων και μεταβάσεων. Σε κάθε κατάσταση κρατάμε την μέγιστη πιθανότητα.  
Αφού τις υπολογίσουμε όλες, με την μέθοδο του backtracking, θα βρούμε τις δυο βέλτιστες ακολουθίες καταστάσεων για την ακολουθία ρίψεων ζαριών : **112122.**

## Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-11\_6.py**.

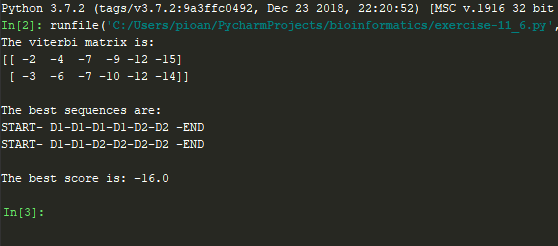
Με βάση τα δεδομένα μας, οι δυο βέλτιστες ακολουθίες που προκύπτουν, συμπεριλαμβανομένων των καταστάσεων START και END είναι οι:

1. **START-D1-D1-D1-D1-D2-D2-END**
2. **START-D1-D1-D2-D2-D2-D2-END**

Ο Viterbi πίνακας που περιλαμβάνει τις μέγιστες πιθανότητες προέκυψαν είναι:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D1 | -2 | -4 | -7 | -9 | -12 | -15 |
| D2 | -3 | -6 | -7 | -10 | -12 | -14 |

Στην εικόνα 7.1 παρουσιάζεται το αποτέλεσμα της εκτέλεσης του προγράμματος. Εκτός από τις δυο βέλτιστες ακολουθίες καταστάσεων παρουσιάζονται και οι μέγιστες πιθανότητες για κάθε κόμβο καθώς επίσης και το βέλτιστο Viterbi score to οποίο συμπεριλαμβανομένων του αρχικού και τελικού κόμβου είναι -16.



Εικόνα 7.1

## Αναφορές

* Εισαγωγή στους αλγόριθμους Βιοπληροφορικής

Neil. C Jones, Pavel A. Pevzner

* Αναγνώριση προτύπων

Σ. Θεοδωρίδης, Κ. Κουτρούμπας

* Σημειώσεις Αναγνώριση Προτύπων, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά

Πικράκης Άγγελος, PhD

* <https://en.wikipedia.org/wiki/Viterbi_algorithm>

# Βοηθητικά αρχεία

Η κλάση Data που βρίσκεται στο αρχείο **data.py** περιέχει την στατική μέθοδο load\_data(filename). Αυτή η μέθοδος δέχεται ως όρισμα το όνομα του αρχείου από το οποία θέλουμε να φορτώσουμε κάποια ακολουθία.

Επίσης στο φάκελο **data** βρίσκονται τα αρχεία που περιέχουν τις ακολουθίες που χρησιμοποιήθηκαν για τις ασκήσεις.