1η Εργασία

Αλγόριθμοι και Πολυπλοκότητα

5ο Εξάμηνο

Όνομα: Μαρία

Επώνυμο: Σταυροπούλου

Α.Μ. 2058

Πίνακας περιεχομένων

1. Υλικό που χρησιμοποιήθηκε για τα πειράματα

Το υλικό που χρησιμοποιήθηκε για τα πειράματα είναι:

- Επεξεργαστής: Intel(R) Core(TM) i3-4000M CPU @ 2.40GHz
- Εγκατεστημένη RAM: 4,00 GB
- Τύπος συστήματος: Λειτουργικό σύστημα 64 bit, επεξεργαστής τεχνολογίας x64

2. Λογισμικό που χρησιμοποιήθηκε

Το λογισμικό που χρησιμοποιήθηκε είναι:

- Python 3.8.10
- PyCharm Community
- Matplotlib

3. Θεωρητική πολυπλοκότητα των αλγορίθμων.

Το πρόβλημα που μας ζητείτε να επιλύσουμε είναι το πρόβλημα της μέγιστης υποακολουθίας (maximum subarray). Ένα πρόβλημα στο οποίο δίνεται μια ακολουθία τιμών (αρνητικές, θετικές ή μηδέν) και ζητείται να βρεθεί η συνεχόμενη σειρά τιμών της ακολουθίας που δίνει το μεγαλύτερο δυνατό άθροισμα. Στην παρούσα εργασία ζητείται η αλγοριθμική διερεύνηση του προβλήματος. Επιπλέον, ζητείται η υλοποίηση αλγορίθμων επίλυσης του προβλήματος με διαφορετική υπολογιστική πολυπλοκότητα, η καταγραφή και η παρουσίαση των χρόνων εκτέλεσης.

a) Ο πρώτος αλγόριθμος είναι ο απλοϊκός τρόπος επίλυσης του προβλήματος. Υλοποιείται χρησιμοποιώντας 3 εμφωλευμένους βρόχους επανάληψης και έναν αθροιστή. Αν και πιο απλοϊκός δεν είναι και ο πιο

γρήγορος, διότι λόγω των εμφωλευμένων βρόχων επανάληψης η πολυπλοκότητα του αλγορίθμου θα είναι: $O(n^3)$.

b) Ο δεύτερος αλγόριθμος είναι ένας πιο βελτιωμένος και πιο γρήγορος αλγόριθμος σε σχέση με τον προηγούμενο. Ο προηγούμενος τρόπος “χάνει” χρόνο καθώς υπολογίζει ξανά όλα τα αθροίσματα υποακολουθίας. Ο αλγόριθμος μπορεί να βελτιωθεί υπολογίζοντας εκ των προτέρων όλα τα προθεματικά αθροίσματα της ακολουθίας. Διαθέτοντας τα προθεματικά αθροίσματα μπορούμε να υπολογίσουμε το άθροισμα οποιασδήποτε υποακολουθίας πραγματοποιώντας αφαίρεση ανάμεσα σε δύο ήδη υπολογισμένες τιμές προθεματικών αθροισμάτων. Για να γίνει αυτή η αφαίρεση στον αλγόριθμο θα έχουμε 2 εμφωλευμένους βρόχους επανάληψης άρα η πολυπλοκότητα του αλγορίθμου θα είναι $O(n^2)$.

c) Ο τρίτος αλγόριθμος λέγεται και αλγόριθμος του Kadane. Η βασική ιδέα του αλγορίθμου του Kadane είναι ότι υπολογίζει μέγιστα επιθεματικά αθροίσματα μηδενίζοντας όσα αθροίσματα από αυτά που προκύπτουν είναι αρνητικοί αριθμοί. Αυτό το επιτυγχάνει με την βοήθεια ενός βρόγχου επανάληψης. Άρα η πολυπλοκότητα αυτού του αλγορίθμου θα είναι $O(n)$

4. Μετρήσεις χρόνου εκτέλεσης του αλγορίθμου.

Τους αλγόριθμους τους έτρεξα με τυχαία δεδομένα εισόδου 10, 100, 1.000, 5.000 ακεραίων τιμών σε εύρος τιμών από -100 έως και 100.

Οι χρόνοι εκτέλεσης των αλγορίθμων αποτυπώνονται στον παρακάτω πίνακα (επειδή ο κώδικας που έχω φτιάξει βγάζει διαφορετικές τυχαίες τιμές κάθε φορά οι παρακάτω είναι από όταν το έτρεξα για να εμφανιστεί το διάγραμμα που του επόμενου ερωτήματος)

	First Algorithm $O(n^3)$	Second Algorithm $O(n^2)$	Third Algorithm $O(n)$
10	0.0001049000000001854 seconds	1.900000006571645e-05 seconds	1.0700000075303251e-05 seconds
100	0.02914249999999985 seconds	0.0007718000001659675 seconds	1.4600000213249587e-05 seconds
1000	27.0943984 seconds	0.1272662000001219 seconds	0.0004020999999738706 seconds
5000	2392.0712719999997 seconds	2.843735000000379 seconds	0.0006630000002587622 seconds

5. Γραφική σύγκριση μέσω διαγραμμάτων των αποτελεσμάτων που παράγουν οι αλγόριθμοι.