Πανεπιστήμιο Ιωαννίνων Τμήμα Πληροφορικής και Τηλεπικοινωνίων Αλγόριθμοι και πολυπλοκότητα

To πρόβλημα της μέγιστης υποακολουθίας

Κώστας Άγγελος

2 Νοεμβρίου 2022

## Περίληψη

Με τον όρο πολυπλοκότητα ορίζουμε την διαδικασία υπολογισμού ενός τύπου ο οποίος προσδιορίζει τον συνολικό χρόνο που απαιτείται για την εκτέλεση ενός συγκεκριμένου αλγορίθμου για είσοδο δεδομένων μεγέθους (n). Στην συγκεκριμένη εργασία αναλύονται τρείς αλγόριθμοι που λύνουν το πρόβλημα της μέγιστης υποακολουθίας. Για κάθε έναν ξεχωριστά υπολογίζουμε την πολυπλοκότητα, τονίζοντας έτσι την διαφορά σε ταχύτητα λόγο της επιλογής ενός ικανότερου αλγόριθμου.

## 1. Εισαγωγή

Το πρόβλημα ουσιαστικά είναι το εξής : Δεδομένου μια ακολουθία ακεραίων τιμών μας ζητείται να βρούμε την υποακολουθία με το μεγαλύτερο άθροισμα. Στην συγκεκριμένη αναφορά παρουσιάζονται τρείς αλγόριθμοι που λύνουν το πρόβλημα :

1) Brute Force Algorithm

Αποτελεί τον πιο απλό τρόπο υπολογισμού της υποακολουθίας γι’ αυτό και έχει πολυπλοκότητα ίση με *Ο()*.

2) Prefix Algorithm

Αποτελεί μια καλύτερη προσέγγιση με πολυπλοκότητα ίση με *Ο()*.

3) Kadane Algorithm

Αποτελεί τον πιο γρήγορο τρόπο επίλυσης του προβλήματος με πολυπλοκότητα ίση με *Ο(n)*.

## 2. Πειραματικά αποτελέσματα

Για την καταγραφή των πειραματικών αποτελεσμάτων χρησιμοποιήθηκε το εξής setup :

* IDE : PyCharm 2022.2.3
* Γλώσσα προγραμματισμού : Python 3.10.8

Χαρακτηριστικά υπολογιστή :

* CPU : AMD Ryzen 5 3400g
* RAM : 32 GB DDR4 (2933 MHz)
* Βιβλιοθήκη γραφικών : matplotlib

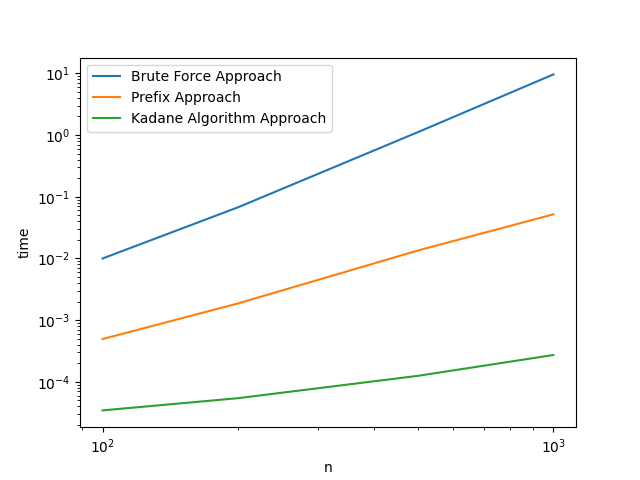
Παράμετροι πειραμάτων :

Για την μέτρηση του χρόνου εκτέλεσης των αλγορίθμων χρησιμοποιήθηκαν τυχαίοι αριθμοί με εύρος το [-100,100]. Για κάθε έναν αλγόριθμο χρησιμοποιήθηκε πλήθος από διάφορα *N*. Στην συγκεκριμένη αναφορά έχουμε: .

Παρακάτω παρουσιάζονται τα αποτελέσματα του χρόνου εκτέλεσης για διαφορετικό αριθμό από *Ν*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **N=100** | **N=1.000** | **N=5.000** | **N=10.000** |
| **Brute Force Algorithm** | 0.0099814 | 0.06766540 | 1.110038499 | 9.59596 |
| **Prefix Algorithm** | 0.000499 | 0.0018724 | 0.0134182 | 0.0519 |
| **Kadane Algorithm** | 3.449999 | 5.45000 | 0.00012 | 0.00027 |

Πίνακας . Πίνακας αποτελεσμάτων



Εικόνα . Οπτικοποίηση αποτελεσμάτων ( matplotlib)

## 2. Συμπεράσματα

Από το συγκεκριμένο πρόβλημα αλλά και τις μετρήσεις που εκτελέσαμε φαίνεται η χρησιμότητα σε χρόνο και χώρο από την επιλογή ενός ικανού και γρηγορότερου αλγόριθμου σε σχέση με απλούστερες υλοποιήσεις.