Πρόβλημα μέγιστου αθροίσματος υποακολουθίας

ΟΡΙΣΜΟΣ: Στην πληροφορική το πρόβλημα μέγιστου αθροίσματος υποακολουθίας (Maximum subarray problem) είναι ένα πρόβλημα όπου έχουμε μια σειρά (πίνακα) από θετικούς και αρνητικούς αριθμούς και ψάχνουμε την συνεχή μεγαλύτερη υποσειρά (υποπίνακα) αριθμών όπου έχουμε το μεγαλύτερο άθροισμα των στοιχείων.

ΑΛΓΟΡΙΣΘΜΟΣ ΜΕ ΤΑΞΗ ΣΥΓΚΛΙΣΗΣ O(n^3):

Αυτός ο αλγόριθμος βασίζεται στην ιδέα ότι για να αντιμετωπίσω το συγκεκριμένο πρόβλημα μπορώ να βρω όλους τους πιθανούς υποπίνακες (2 loops) και όλα τα πιθανά αθροίσματα και να τα συγκρίνω μεταξύ τους (Τρίτο εσωτερικό loop) . Άρα υπάρχουν τρία εσωτερικά loop το ένα μέσα στο άλλο που εκτελούν n επαναλήψεις το καθένα. Άρα η τάξη σύγκλισης είναι O(n^3).

ΑΛΓΟΡΙΣΘΜΟΣ ΜΕ ΤΑΞΗ ΣΥΓΚΛΙΣΗΣ O(n^2):

Αυτός ο αλγόριθμος είναι παρόμοιας λογικής με τον προηγούμενο με μια διαφορά. Για να υπολογίσει το άθροισμα από A(j)…A(j+1) προσθέτει το A(j+1) στο άθροισμα A(j)…A(j). Με αυτό τον τρόπο αποφεύγουμε το τρίτο εσωτερικό loop. Άρα υπολογίζεται ότι η τάξη σύγκλισης είναι O(n^2).

ΑΛΓΟΡΙΣΘΜΟΣ ΜΕ ΤΑΞΗ ΣΥΓΚΛΙΣΗΣ O(nlogn):

Αυτός ο αλγόριθμος (Διαίρει και Βασίλευε) βασίζεται στην ιδέα ότι για να λύσω ένα πρόβλημα το χωρίζω σε υποπροβλήματα. Στο συγκεκριμένο πρόβλημα αρχικά διαιρώ τον πίνακα σε δύο υποπίνακες. Τρώα ο υποπίνακας με το μεγαλύτερο άθροισμα μπορεί να βρίσκεται είτε μόνο στον αριστερό υποπίνακα, είτε

μόνο στον δεξιό υποπινακα, είτε να περιλαμβάνει στοιχεία και των δύο. Ανάλυση σύγκλισης:

Για κάθε επιμέρους βήμα πρέπει να υπολογίζω τη μέγιστη υποσειρα προχωρώντας μια φορά κατά O(n).

Έτσι:

$T(n) = T(n/2) + T(n/2) + O(n)$ (1)

$T(n) = 2 \times T(n/2) + O(n)$ (2)

$T(n) = 2 \times \{2 \times T(n/4) + O(n/2)\} + O(n)$ (3)

$T(n) = 4 \times T(n/4) + 2 \times O(n)$ (4)

… = … (5)

$T(n) = n \times T(n/n) + \log n \times O(n)$ (6)

$T(n) = O(n\log n)$ (7)

<u>ΑΛΓΟΡΙΣΘΜΟΣ ΜΕ ΤΑΞΗ ΣΥΓΚΛΙΣΗΣ O(n):</u>

Αυτός ο αλγόριθμος έχει παρόμοια λογική με τον προηγούμενο, δηλαδή διαιρεί το πρόβλημα σε υποπρόβληματα. Σύμφωνα με αυτόν σαρώνουμε τις τιμές του πίνακα, και κάθε φορά υπολογίζουμε την θέση με το μέγιστο θετικό άθροισμα της υποσειράς που τερματίζεται στο συγκεκριμένο σημείο. Έχει ένα loop που εκτελεί n επαναλήψεις.

ΔΕΔΟΜΕΝΑ 1: 500 ΣΤΟΙΧΕΙΑ

ΔΕΔΟΜΕΝΑ 1: 1000 ΣΤΟΙΧΕΙΑ

ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΕΔΟΜΕΝΑ1 ΧΡΟΝΟΣ	ΔΕΔΟΜΕΝΑ2 ΧΡΟΝΟΣ
n^3	4.758030499999999	40.9814627
n^2(χωρίς πίνακα)	0.03120020000000001	0.12480080000000002
n^2(με πίνακα)	0.7176046	5.2260335
nlogn	0.0	0.015600100000000006
n	0.0	0.0

Από τον παραπάνω πίνακα επιβεβαιώνεται ότι ο αλγόριθμος με σύγκλιση n (dynamic programming) είναι ο γρηγορότερος. Ακολουθούν κατά σειρά nlogn, n^2 χωρίς πίνακα, n^2 με πίνακα, και τελευταίος ο n^3.