

Συνάρτηση sort

Η παρεχόμενη συνάρτηση ταξινόμησης υλοποιεί μια παραλλαγή του αλγορίθμου ταξινόμησης εισαγωγής.

Αρχικός βρόχος σύγκρισης:

```
για (int i = l+1; i <= r; i++) {  
    αν (a[i] < a[l]) {  
        Στοιχείο t = a[i];  
        a[i] = a[l];  
        a[l] = t;  
    }  
}
```

Αυτός ο βρόχος εκτείνεται από το $l+1$ έως το r και συγκρίνει κάθε στοιχείο με το πρώτο στοιχείο ($a[l]$). Εάν κάποιο στοιχείο είναι μικρότερο από το πρώτο στοιχείο, ανταλλάσσονται. Αυτός ο βρόχος διασφαλίζει ότι το μικρότερο στοιχείο στο τμήμα πίνακα $[l, r]$ μετακινείται στη θέση l . Ο βρόχος εκτελείται για επαναλήψεις $r-l$, που είναι $O(n)$ όπου n είναι ο αριθμός των στοιχείων στο τμήμα.

Βρόχος ταξινόμησης εισαγωγής:

```
για (int i = l+2; i <= r; i++) {  
    int j = i;  
    Στοιχείο v = a[i];  
    ενώ (v < a[j-1]) {  
        a[j] = a[j-1];  
        j--;  
    }  
    a[j] = v;  
}
```

Αυτός είναι ο κύριος βρόχος της ταξινόμησης εισαγωγής, ξεκινώντας από το $l+2$. Για κάθε στοιχείο, βρίσκει τη σωστή θέση συγκρίνοντάς το προς τα πίσω μέχρι να φτάσει στην αρχή του πίνακα ή να βρει ένα στοιχείο μικρότερο από το ίδιο. Το χειρότερο σενάριο συμβαίνει όταν ο πίνακας βρίσκεται σε αντίστροφη σειρά, με αποτέλεσμα κάθε στοιχείο να συγκρίνεται δυνητικά με όλα τα προηγούμενα στοιχεία. Ο αριθμός των συγκρίσεων και των μετατοπίσεων στη χειρότερη περίπτωση είναι περίπου $n(n-1)/2$ (όπου n είναι ο συνολικός αριθμός των στοιχείων), κάνοντας αυτόν τον βρόχο $O(n^2)$ στη χειρότερη περίπτωση.

Συνάρτηση υπολογιστικής πολυπλοκότητας ταξινόμησης:

Δεδομένων των παραπάνω βρόχων, ο πρώτος βρόχος εκτελείται σε χρόνο $O(n)$, αλλά ο δεύτερος βρόχος, ο οποίος κυριαρχεί, εκτελείται σε χρόνο $O(n^2)$ στη χειρότερη περίπτωση. Επομένως, η χρονική πολυπλοκότητα στη χειρότερη περίπτωση της συνάρτησης ταξινόμησης είναι $O(n^2)$.

Συνάρτηση main

Η πολυπλοκότητα της συνάρτησης main περιλαμβάνει:

Δημιουργία τυχαίων αριθμών και συμπλήρωση του πίνακα:

```
για (int i = 0; i < N; i++)
```

```
a[i] = 1000*(1,0*rand()/RAND_MAX); // [0, 1000]
```

Αυτός ο βρόχος εκτελείται N φορές, και κάθε λειτουργία μέσα σε αυτόν (δημιουργώντας έναν τυχαίο αριθμό και εκχωρώντας τον) είναι $O(1)$. Επομένως, αυτό το τμήμα είναι $O(N)$.

Ταξινόμηση:

Η συνάρτηση ταξινόμησης καλείται μία φορά, και όπως αναλύθηκε παραπάνω, εκτελείται σε $O(n^2)$ στη χειρότερη περίπτωση.

Εκτύπωση του πίνακα (δύο φορές):

Κάθε λειτουργία εκτύπωσης εκτελείται N φορές (μία πριν από την ταξινόμηση και μία μετά), κάθε εκτύπωση είναι μια λειτουργία $O(1)$. Έτσι, κάθε βρόχος για εκτύπωση είναι $O(N)$, και αν το κάνετε δύο φορές το κάνει $2 \cdot O(N) = O(N)$.

Υπολογιστική πολυπλοκότητα της συνάρτησης main:
Συνδυάζοντας τις πολυπλοκότητες, παίρνουμε:

Αρχικός πληθυσμός: $O(N)$

Ταξινόμηση: $O(n^2)$

Εκτύπωση: $O(N)$

Ο κυρίαρχος παράγοντας εδώ είναι η λειτουργία ταξινόμησης, η οποία είναι $O(n^2)$. Έτσι, η συνολική χρονική πολυπλοκότητα στη χειρότερη περίπτωση της κύριας συνάρτησης είναι $O(n^2)$, που υπαγορεύεται κυρίως από τη λειτουργία ταξινόμησης.

Τόσο η συνάρτηση ταξινόμησης όσο και η κύρια συνάρτηση παρουσιάζουν μια χρονική πολυπλοκότητα στη χειρότερη περίπτωση $O(n^2)$, όπου n είναι ο αριθμός των στοιχείων που ταξινομούνται. Η πολυπλοκότητα της κύριας συνάρτησης επηρεάζεται ιδιαίτερα από τη λειτουργία ταξινόμησης, παρόλο που άλλες λειτουργίες (όπως η δημιουργία αριθμών και η εκτύπωση) έχουν γραμμική πολυπλοκότητα.