ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

Ενότητα 8β: Ταξινόμηση-Ταξινόμηση του Shell

Μαρία Σατρατζέμη
Τμήμα Εφαρμοσμένης Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Ταξινόμηση του Shell

- Η μέθοδος που προτάθηκε από τον Shell έχει βασικό χαρακτηριστικό ότι χρησιμοποιεί μία ακολουθία ακεραίων $h_1, h_2, \ldots, h_k$, όπου ισχύει:
  $h_1 > h_2 > \ldots > h_k = 1$
  και γι' αυτό φέρει και την ονομασία **ταξινόμηση με μειούμενες αυξήσεις** (*diminishing increment*).
- Στην πράξη η μέθοδος λειτουργεί για οποιεσδήποτε τιμές της ακολουθίας $h_1, h_2, \ldots, h_{k-1}$ αλλά θα πρέπει να ισχύει $h_k = 1$.

- Η ταξινόμηση του Shell είναι μία παραλλαγή της ταξινόμησης με εισαγωγή.
- Η μέθοδος αποτελείται από k φάσεις, όπου στην i-οστή φάση (για $1 \leq i \leq k$) θεωρείται το βήμα h_i, οπότε ακολουθώντας τη λογική της ταξινόμησης με εισαγωγή τίθενται σε σωστή διάταξη μεταξύ τους τα στοιχεία του πίνακα που απέχουν h_i θέσεις.
- Προφανώς η k-οστη και τελευταία φάση είναι ένα κλασικό πέρασμα της ταξινόμησης με εισαγωγή που ολοκληρώνει τη διαδικασία.
- Αν ο πίνακας είναι ταξινομημένος με βάση κάποιο βήμα, τότε παραμένει ταξινομημένος με βάση το ίδιο βήμα ακόμη και αν εφαρμοσθούν ένα ή περισσότερα επόμενα βήματα στη συνέχεια.
- Επίσης, καθώς η μέθοδος αυτή στηρίζεται στην ταξινόμηση με εισαγωγή, είναι και αυτή ευσταθής, δηλαδή δεν ανταλλάσσει αμοιβαία στοιχεία με ίσα κλειδιά.

Αλγόριθμος Shell

```
h = ⌊n/2⌋;
while (h >0) do
    for i ← h to n do
        tmp ← A[i]; j ← i;
        while ((j > h and A[j - h] > tmp ) do
            A[j] ← A[j-h]; j ← j – h;
        A[j] ← tmp
    h = ⌊h/2⌋;
```

Ένα παράδειγμα

με 9 κλειδιά, ταξινόμηση με 3 βήματα μεγέθους 4, 2, 1

Αρχικά Κλειδιά	52	12	71	56	5	10	19	90	45
	5				52				
					45				52
		10				12			
			19				71		
Βήμα 4	5	10	19	56	45	12	71	90	52
							52		71
				12		56			
Βήμα 2	5	10	19	12	45	56	52	90	71
			12	19					
						52	56		
								71	90
Βήμα 1	5	10	12	19	45	52	56	71	90

Δες το <u>Shell_Srt_Example.pdf</u>

Πρόταση.

Η πολυπλοκότητα της shellsort είναι Θ(n^2) στη χειρότερη περίπτωση, για την ακολουθία του Shell, όπου $h_i = \lfloor n/2 \rfloor$ *και* $h_{i+1} = \lfloor h_i/2 \rfloor$

Απόδειξη

Για να αποδείξουμε ότι η πολυπλοκότητα της χειρότερης περίπτωσης είναι *Θ(n^2),* πρώτα θα αποδείξουμε ότι ισχύει το Ω(n^2) και μετά το *O(n^2).*

Αποδεικνύουμε το κάτω φράγμα πρώτα κατασκευάζοντας μια «κακή» περίπτωση.

Έστω ότι $n = 2^k$, οπότε κάθε βήμα είναι άρτιος αριθμός εκτός ató το τελευταίο βήμα που ισούται με τη μονάδα.

Θεωρούμε τα δεδομένα εισόδου αρχικά τοποθετούνται τα *n/2* μεγαλύτερα στοιχεία στις άρτιες θέσεις, ενώ τα *n/2* μικρότερα στοιχεία τοποθετούνται στις περιττές θέσεις. Καθώς όλα τα βήματα είναι άρτιοι αριθμοί εκτός από το τελευταίο που είναι περιττός όταν θα φθάσουμε στο τελευταίο βήμα, όλα τα *n/2* μεγαλύτερα στοιχεία είναι ακόμη στις άρτιες θέσεις και όλα τα *n/2 μικρότερα* στοιχεία είναι στις περιττές θέσεις.

Επομένως το *i*-οστό μικρότερο στοιχείο (για $i \le n/2$) βρίσκεται στην (2*i* - 1)-οστή θέση πριν αρχίσει το τελικό πέρασμα και πρέπει να εκτελεσθούν *i* - 1 ανταλλαγές μέχρι να φθάσει στην τελική του θέση. Άρα, για όλα τα *n/2* μικρότερα στοιχεία θα απαιτηθούν

$$\sum_{i=1}^{n/2}(i-1) = \frac{\frac{n}{2}\left(\frac{n}{2}-1\right)}{2} = \frac{n^2-2n}{8} = \Omega\left(n^2\right) \qquad (1)$$

ανταλλαγές

Start	1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
After 8-sort	1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
After 4-sort	1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
After 2-sort	1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
After 1-sort	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Ένα κακό παράδειγμα αρχικών δεδομένων (όχι το χειρότερο). Το πλήθος των ανταλλαγών μετά το 2-sort είναι 1+2+3+4+5+6+7=28, και προφανώς το τελευταίο πέρασμα θα απαιτήσει σημαντικό χρόνο.

Δεύτερο σκέλος

Αρκεί να θεωρήσουμε ότι κατά το πέρασμα με βήμα h_k, ουσιαστικά εκτελείται μία ταξινόμηση με εισαγωγή (τετραγωνικής πολυπλοκότητας) για ένα σύνολο n/h_k στοιχείων.

Επομένως, για το σύνολο των περασμάτων με αυτό το βήμα η πολυπλοκότητα είναι $O\left(h_k (n/h_k)^2\right) = O\left(n^2/h_k\right)$

Επομένως για το σύνολο των βημάτων προκύπτει η πολυπλοκότητα

$$O\left(\sum_{i=1}^{t} n^2/h_t\right) = O\left(n^2 \sum_{i=1}^{t} 1/h_t\right)$$

Καθώς στη γεωμετρική σειρά $\sum_{i=1}^{t} 1/h_t$ (καθώς $n = 2^k$ $h_i = \lfloor n/2 \rfloor$ $h_{i+1} = \lfloor h_i/2 \rfloor$) ο μεγαλύτερος όρος είναι h_1=1, τότε το άθροισμα των n όρων γεωμετρικής σειράς είναι $O(t(n))$, όπου $t(n)$ ο μεγαλύτερος όρος του αθροίσματος (δες πρόταση σελ. 5, 04_Recurrence Relations.pdf), άρα $O(1)$. Συνεπώς

$$O\left(\sum_{i=1}^{t} n^2/h_t\right) = O\left(n^2 \sum_{i=1}^{t} 1/h_t\right) = O\left(n^2\right) \qquad (2)$$

Η πολυπλοκότητα της ταξινόμησης του Shell για δεδομένη ακολουθία βημάτων από τις (1) & (2) προκύπτει ότι είναι στη χειρότερη περίπτωση $\Theta\left(n^2\right)$

Εναλλακτικές προτάσεις στην ακολουθία των βημάτων

Στη βιβλιογραφία έχουν παρουσιασθεί πολλές εναλλακτικές προτάσεις για την ακολουθία των βημάτων και έχουν αναλυθεί. Για παράδειγμα, για την ακολουθία 1, 3, 7, ... , 2^k - 1 του Hibbard, αν και εικάζεται από πειραματικά αποτελέσματα ότι η πολυπλοκότητα είναι $O\left(n^{5/4}\right)$ το όριο που έχει αποδειχθεί είναι $O\left(n^{3/2}\right)$

Ο Sedgewick πρότεινε αρκετές ακολουθίες βημάτων που δίνουν πολυπλοκότητα χειρότερη περίπτωσης $O\left(n^{4/3}\right)$

Η πολυπλοκότητα μέσης περίπτωσης εικάζεται ότι είναι $O\left(n^{7/6}\right)$

Η καλύτερη ακολουθία βημάτων που απέδειξε ο Sedgewick είναι η 1, 5, 19, 41, 109, ….

Ο αλγόριθμος του Shell είναι περίπτωση ενός απλού αλγόριθμου με ιδιαίτερα απαιτητικές τις αποδείξεις πολυπλοκότητας.

Στην πράξη δίνει καλά αποτελέσματα για τιμές του *n* στις δεκάδες χιλιάδες, και επιλέγεται για σχετικά μεγάλα μεγέθη δεδομένων λόγω της απλότητας του κώδικα

Μια αναλυτικότερη παρουσίαση του shellsort δείτε http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/shell/shellen.htm

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ