

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ**

**1<sup>η</sup> άσκηση**

**Ημερομηνία παράδοσης:** 20 Μαρτίου 2019

**Η άσκηση είναι ατομική**

**Μέρος 1<sup>ο</sup>: Επεξεργασία Αρχείων - Εξωτερική ταξινόμηση**

Το αρχείο ευρίσκεται στον δίσκο. Η επεξεργασία των στοιχείων γίνεται στην κεντρική μνήμη διαβάζοντας από το αρχείο 1 buffer μεγέθους 4Kbytes κάθε φορά. Το μέγεθος του buffer αντιστοιχεί στο μέγεθος σελίδας του αρχείου. Αντίστοιχα, εγγραφή στο αρχείο γίνεται εγγράφοντας 1 buffer κάθε φορά (δηλαδή μία σελίδα αρχείου). Ποτέ δεν γίνεται ανάγνωση ή εγγραφή στοιχείο προς στοιχείο δηλαδή για μέγεθος μικρότερο από το μέγεθος του buffer. Αντίστοιχα δεν μπορεί να γίνει ανάγνωση ή εγγραφή στο αρχείο με μέγεθος μεγαλύτερο ενός buffer (δηλαδή μίας σελίδας αρχείου). Εφόσον το αρχείο αποθηκεύει ακέραιους αριθμούς σε δυαδική μορφή, το buffer έχει 1.000 αριθμούς. Κάθε ανάγνωση ή εγγραφή στον δίσκο αυξάνει τον αριθμό προσβάσεων στον δίσκο κατά 1.

**Μέρος 1ο: Κατασκευή αρχείου N αριθμών (20 μονάδες)**

Δημιουργήστε ένα αρχείο με  $N = 10^5$  τυχαίους ακέραιους αριθμούς στο δίσκο. Το αρχείο περιέχει τυχαίους ακέραιους αριθμούς που παράγονται από μια γεννήτρια τυχαίων αριθμών. Κάθε αριθμός παριστάνεται σε δυαδική μορφή 4 Bytes (το αρχείο είναι σε δυαδική μορφή και όχι text).

Για την κατασκευή του αρχείου εφαρμόστε μια επαναληπτική διαδικασία:

1. Γεμίστε το buffer με 1.000 τυχαίους αριθμούς
2. Γράψτε το buffer στο αρχείο (αυτή είναι η 1<sup>η</sup> σελίδα στο αρχείο και απαιτεί μία πρόσβαση στον δίσκο)
3. Επαναλάβετε τα βήματα 1 και 2 μέχρι να δημιουργηθεί ένα αρχείο  $N = 10^5$  αριθμών (γράφοντας στο αρχείο την επόμενη σελίδα μετά την προηγούμενη – ταυτόχρονα αυξάνεται ο αριθμός των προσβάσεων στον δίσκο).

Κάθε εγγραφή σελίδας (δηλαδή του buffer) στο αρχείο μετράει ως μία πρόσβαση στο δίσκο. Μετρήστε πόσες προσβάσεις δίσκου θα χρειαστούν για την κατασκευή του αρχείου και επιβεβαιώστε ότι είναι 100. Αυτός είναι και ο αριθμός σελίδων στο αρχείο.

**Μέρος 2<sup>ο</sup>: Ταξινόμηση του αρχείου στον δίσκο (20 μονάδες)**

Θα χρειαστείτε 1 buffer μεγέθους 1.000 αριθμών όπως και στο προηγούμενο ερώτημα. Ταξινομήστε το αρχείο στο δίσκο με την παρακάτω μέθοδο:

1. Διαβάστε 10.000 αριθμούς από το αρχείο στην κεντρική μνήμη (διαβάζοντας διαδοχικά 10 σελίδες από το αρχείο). Αυτό σημαίνει 10 διαδοχικές προσβάσεις δίσκου. Τοποθετήστε τους 10.000 αριθμούς σε ένα πίνακα (array).
2. Ταξινομήστε το πίνακα με μέθοδο quicksort ή mergesort ή heapsort (βρείτε έτοιμη υλοποίηση από το web).
3. Γράψτε τον ταξινομημένο πίνακα στο δίσκο σε ένα νέο ταξινομημένο αρχείο. Αυτό σημαίνει 10 διαδοχικές εγγραφές buffer ή 10 προσβάσεις δίσκου.

4. Επαναλάβετε τα βήματα 1 έως 3 συνεχίζοντας το διάβασμα του αρχείου διαβάζοντας τους επόμενους 10.000 αριθμούς και δημιουργώντας νέα ταξινομημένο αρχείο.
5. Στο τέλος δημιουργούνται στο δίσκο 10 ταξινομημένα αρχεία μεγέθους 10.000 αριθμών το καθένα. Επιβεβαιώστε ότι χρειάστηκαν 200 προσβάσεις στο δίσκο.

Στην συνέχεια θα γίνει «συγχώνευση των 10 ταξινομημένων» αρχείων σε 1 τελικό αρχείο.

**Απαγορεύεται να κρατήσετε και τα 10 αρχεία στην κεντρική μνήμη ταυτόχρονα.** Θα χρειαστείτε 11 buffers μεγέθους 1.000 αριθμών ο καθένας (μπορείτε να έχετε ένα array από 11 buffers δηλαδή arrays 1.000 ακεραίων). Κάθε φορά που διαβάζετε ένα buffer από τον δίσκο ή γράφετε έναν buffer στον δίσκο θα μετράτε 1 νέα πρόσβαση στον δίσκο.

1. Ο καθένας από τους 10 buffer διαβάζει τους πρώτους 1.000 αριθμούς από καθένα από τα 10 αρχεία (ένas buffer ανά αρχείο). Δηλαδή, από κάθε ταξινομημένο αρχείο διαβάστε 1 buffer (1.000 αριθμούς). Έτσι θα γεμίσετε τους πρώτους 10 buffers.
2. Διαβάζετε τον πρώτο αριθμό τους από τον κάθε buffer και γεμίζετε τον 11<sup>ο</sup> buffer γράφοντας τον μικρότερο από τους 10 αριθμούς. Στον buffer που ανήκει ο μικρότερος αριθμός προχωράτε μία θέση (πάτε στον επόμενο αριθμό). Μετά συνεχίζετε γράφοντας κάθε φορά τον επόμενο μικρότερο αριθμό.
3. Συνεχίζεται τα βήματα 1 και 2
4. Κάθε φορά που γεμίζει ο 11<sup>ος</sup> buffer, τον γράφετε σε ένα νέο αρχείο.
5. Κάθε φορά που εξαντλείται οποιοσδήποτε από τους 10 buffers (δηλαδή έχουμε διαβάσει όλα τα στοιχεία του), τον ξαναγεμίζουμε διαβάζοντας τους επόμενους 1.000 αριθμούς (την επόμενη σελίδα) από το αντίστοιχο (από το 10) αρχείο στο δίσκο.
6. Αν ένα αρχείο από τα 10 εξαντληθεί τότε (μπορούμε να το σβήσουμε από τον δίσκο) συνεχίζουμε την διαδικασία με τα υπόλοιπα

Στο τέλος, όλοι οι αριθμοί που διαβάστηκαν από τα 10 αρχεία έχουν μεταφερθεί στο 11<sup>ο</sup> αρχείο που θα είναι το τελικό ταξινομημένο αρχείο. Μετρήστε τον αριθμό των προσβάσεων δίσκου που χρειάστηκε για την ταξινόμηση του αρχείου.

### **Μέρος 3<sup>ο</sup>: Αναζήτηση στο ταξινομημένο αρχείο**

Θα χρειαστεί να κατασκευάσετε 2 μεθόδους για την αναζήτηση ενός τυχαίου αριθμού στο παραπάνω ταξινομημένο αρχείο. Δίνεται ένας αριθμός (λέγεται «κλειδί») και η μέθοδος απαντά αν υπάρχει ή όχι στο αρχείο. Ταυτόχρονα θα μετρήσει τον αριθμό προσβάσεων στον δίσκο που χρειάστηκαν για να απαντήσει. Θα χρειαστεί να κάνετε 40 αναζητήσεις και για κάθε αναζήτηση πρέπει να μετρήσετε τον αριθμό προσβάσεων στον δίσκο. 20 αναζητήσεις θα γίνουν για τυχαίες τιμές κλειδιών που υπάρχουν στο αρχείο και 20 αναζητήσεις για τυχαίες τιμές που δεν υπάρχουν.

1. **Σειριακή αναζήτηση (20 μονάδες):** Για την αναζήτηση, διαβάζετε στην κεντρική μνήμη τα δεδομένα του αρχείου διαβάζοντας 1 buffer κάθε φορά (δηλαδή μία σελίδα αρχείου κάθε φορά). Στην κεντρική μνήμη εξετάζουμε αν υπάρχει ο αριθμός που ψάχνουμε. Αν υπάρχει, τότε έχουμε «επιτυχημένη αναζήτηση». Τότε σταματάμε την διαδικασία και γράφουμε στην έξοδο πόσες προσβάσεις δίσκου χρειάστηκαν. Αν όχι, συνεχίζουμε διαβάζοντας τους επόμενους 1000 αριθμούς (την επόμενη σελίδα του αρχείου) στην κεντρική μνήμη στο buffer. Αν εξαντλήσουμε το αρχείο χωρίς να βρούμε τον αριθμό που ψάχνουμε τότε έχουμε «αποτυχημένη αναζήτηση» και μετράμε πόσες προσβάσεις δίσκου χρειάστηκαν. Για 40 αναζητήσεις τυχαίων αριθμών μετρήστε τον μέσο αριθμό προσβάσεων δίσκου που χρειάστηκαν για επιτυχημένη και αποτυχημένη αναζήτηση αντίστοιχα.
2. **Δυαδική αναζήτηση (20 μονάδες):** Διαβάστε αρχικά την μεσαία σελίδα του αρχείου στην κεντρική μνήμη (δηλαδή στον buffer). Στην κεντρική μνήμη, εξετάστε αν ο αριθμός που ψάχνετε είναι στον buffer. Αν η απάντηση είναι θετική, έχουμε επιτυχημένη αναζήτηση, σταματάμε την διαδικασία και αναφέρουμε πόσες προσβάσεις δίσκου χρειάστηκαν. Αν η απάντηση είναι αρνητική τότε (α) Αν ο αριθμός είναι μικρότερος από τον μικρότερο

αριθμό της σελίδας, τότε φέρτε στην κεντρική την μεσαία σελίδα του αριστερού μισού του αρχείου – αυξάνοντας κατά 1 τον αριθμό προσβάσεων στον δίσκο (γ) Αν ο αριθμός είναι μεγαλύτερος από τον μεγαλύτερο αριθμό της σελίδας, κάντε το ίδιο με την κεντρική σελίδα του δεξιού μισού του αρχείου - αυξάνουμε κατά 1 τον αριθμό προσβάσεων στον δίσκο. Αν εξαντληθούν όλες οι σελίδες του αρχείου χωρίς να βρούμε τον αριθμό που ψάχνουμε, τότε έχουμε αποτυχημένη αναζήτηση. Για 40 αναζητήσεις αριθμών μετρήστε τον μέσο αριθμό προσβάσεων δίσκου που χρειάστηκαν για επιτυχημένη και αποτυχημένη αναζήτηση αντίστοιχα.

#### **Μέρος 4<sup>ο</sup>: Τεκμηρίωση αποτελεσμάτων (20 μονάδες)**

Συγκεντρώστε τα αποτελέσματα όλων των ερωτημάτων στον παρακάτω πίνακα και δικαιολογήστε την απόδοση κάθε μεθόδου. Στην περίπτωση αναζήτησης πρέπει να συγκρίνετε και να αιτιολογήσετε τις αποδόσεις (α) Επιτυχημένης και αποτυχημένης αναζήτησης και (β) Σειριακής και Δυαδικής αναζήτησης. Προσέξτε ότι βαθμολογείται η απάντηση με βάση την τεκμηρίωση (δηλαδή είναι αναμενόμενο το αποτέλεσμα ή όχι και γιατί;)

Μέθοδος	Κατασκευή Αρχείου	Ταξινόμηση Αρχείου	Σειριακή Αναζήτηση (επιτυχημένη)	Σειριακή Αναζήτηση (αποτυχημένη)	Δυαδική Αναζήτηση (επιτυχημένη)	Δυαδική Αναζήτηση (αποτυχημένη)
Απόδοση						

**Παραδοτέα:** Ένα συμπίεσμένο zip αρχείο που περιέχει ότι ζητείται παρακάτω:

- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση.
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πως φτιάχτηκε ο κώδικας (δηλ. για κάθε ερώτημα ποια είναι η γενική ιδέα της λύσης σε 3-4 προτάσεις), υπάρχουν σαφείς οδηγίες μετάφρασης από compiler και εκτέλεσης, τι λάθη έχει (αν έχει, περιπτώσεις που δεν δουλεύει το πρόγραμμα, ή περιπτώσεις που κάνει περισσότερα από όσα σας ζητεί η άσκηση, τι χρησιμοποιήσατε από έτοιμα προγράμματα ή πηγές πληροφόρησης. Υποδείξτε ακόμα και πηγές στο WWW όπως Wikipedia.
- Εκτός των παραπάνω, οι ασκήσεις βαθμολογούνται με άριστα εφόσον:
  - Το zip είναι πλήρες
  - Οι κώδικες περνούν από compiler και εκτελούνται κανονικά και σωστά σε windows ή Linux περιβάλλον
  - Ο κώδικάς σας δουλεύει για οποιοδήποτε ταξινομημένο αρχείο αριθμών που θα δοθεί ως είσοδος.
- Οι ασκήσεις υποβάλλονται ηλεκτρονικά στον ιστοχώρο του μαθήματος και όχι με e-mail
- Οι αντιγραφές μηδενίζονται.