

## Πανεπιστήμιο Πειραιώς – Τμήμα Ψηφιακών Συστημάτων

### Δομές Δεδομένων 2024-2025 – 2η Εργασία\*

**Περιγραφή.** Θα υλοποιήσετε δομή δεδομένων που αναπαριστά *σύνολο*, κάθε στοιχείο του οποίου είναι διατεταγμένο ζεύγος (**id**, **priority**). Για κάθε στοιχείο: **id** είναι μοναδικό αναγνωριστικό, **priority** είναι τιμή προτεραιότητας. Η δομή θα υποστηρίζει με τον *αποδοτικότερο δυνατό τρόπο* τις λειτουργίες:

- εισαγωγής στοιχείου, απομάκρυνσης στοιχείου, απομάκρυνσης στοιχείου μέγιστης προτεραιότητας,
- ένωσης συνόλων, διαφοράς συνόλων, ελέγχου ύπαρξης στοιχείου στη δομή,
- ανάκτησης των  $k$  στοιχείων μεγαλύτερης προτεραιότητας του συνόλου.

Πιο συγκεκριμένα, για σύνολο  $n$  στοιχείων, απαιτείται η πολυπλοκότητα των τριών λειτουργιών: εισαγωγής στοιχείου, απομάκρυνσης στοιχείου, απομάκρυνσης στοιχείου μέγιστης προτεραιότητας να είναι  $O(\log n)$ .

Για την αναπαράσταση συνόλου, ενδείκνυται η δομή πίνακα κατακερματισμού. Στην εργασία, προς υποστήριξη μέρους των παραπάνω πράξεων από τη δομή σας, **ζητείται να ενσωματώσετε πίνακα κατακερματισμού ανοιχτής διευθυνσιοδότησης με δευτεροβάθμια εξέταση**, στον σχεδιασμό και την υλοποίησή της.

**Υπόδειξη:** Να χρησιμοποιηθεί δομή σωρού ίσου μεγέθους με τον πίνακα κατακερματισμού, που να διατηρεί ανά πάσα στιγμή τα ίδια ακριβώς στοιχεία με τον πίνακα κατακερματισμού, οργανωμένα με βάση την προτεραιότητα.

**Μέθοδοι προς υλοποίηση.** Θα επανξήσετε κατάλληλα και θα υλοποιήσετε τις κλάσεις **Node** και **HashHeap** που περιγράφονται παρακάτω. Στην υλοποίησή σας:

- **Επιτρέπεται η τροποποίηση της Node με προσθήκη πεδίων/μεθόδων.**
- **Επιτρέπεται να προσθέσετε στην κλάση HashHeap μόνο *private* πεδία, μεθόδους, ακόμη και ορισμούς κλάσεων (οι οποίες μπορούν να έχουν *public* πεδία/μεθόδους).**<sup>1</sup>
- **Δεν επιτρέπεται χρήση δομών βιβλιοθήκης της Java – δεν επιτρέπεται *import*.**

**Κλάση Node.** Η βασική προδιαγραφή/υλοποίηση της **Node** αναπαριστά ένα στοιχείο (**id**, **priority**):

```
class Node {  
    public int id;  
    public int priority;  
    public Node(int id, int priority)  
}
```

**Κλάση HashHeap.** Ορίζει τη ζητούμενη δομή προς υλοποίηση:

**public HashHeap(int mx)** Κατασκευάζει κενή δομή, με επαρκή χώρο για την αποθήκευση τουλάχιστον μέχρι **mx** στοιχείων. Για να υποστηριχθεί η δευτεροβάθμια εξέταση, και δεδομένης της τιμής εισόδου **mx**, η μέθοδος επιλέγει ως μέγεθος του πίνακα κατακερματισμού **maxSize** έναν πρώτο αριθμό μεγαλύτερο από το διπλάσιο της τιμής **mx**. Η λειτουργικότητα αυτή δίνεται υλοποιημένη με χρήση του πίνακα **primes** που περιέχει ορισμένους πρώτους αριθμούς.

**public boolean insert(Node n)** Επιχειρεί εισαγωγή νέου στοιχείου (**Node**) στη δομή. Εφόσον η εισαγωγή ήταν επιτυχής, επιστρέφει **true**, διαφορετικά επιστρέφει **false**.

**public Node removeId(int id)** Απομακρύνει από τη δομή **this** το στοιχείο με αναγνωριστικό **id**. Επιστρέφει τον κόμβο που αποθηκεύει το στοιχείο αυτό. Επιστρέφει **null**, αν η δομή **this** είναι κενή ή αν το στοιχείο δεν υπάρχει.

\*Χρήστος Δουλκερίδης, Ορέστης Τελέλης

<sup>1</sup> Δείτε π.χ.: <https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html>

`public Node remove()` Απομακρύνει από τη δομή `this` το στοιχείο με μέγιστη προτεραιότητα. Επιστρέφει τον κόμβο που αποθηκεύει το στοιχείο αυτό. Επιστρέφει `null`, αν η δομή `this` είναι κενή.

`public boolean contains(int id)` Επιστρέφει `true`, αν το στοιχείο με αναγνωριστικό `id` περιέχεται στη δομή, διαφορετικά επιστρέφει `false`.

`public HashHeap union(HashHeap h)` Επιστρέφει νέο στιγμιότυπο κλάσης `HashHeap`, που περιέχει την ένωση των στοιχείων της τρέχουσας δομής `this` και της δομής `h`. Κάθε στοιχείο που περιέχεται και στις δύο δομές `this`, `h`, ίσως με διαφορετικές τιμές προτεραιότητας, θα περιέχεται στην επιστρεφόμενη δομή με προτεραιότητα ίση με το άθροισμα των δύο προτεραιοτήτων.

`public HashHeap diff(HashHeap h)` Επιστρέφει νέο στιγμιότυπο κλάσης `HashHeap`, που περιέχει στοιχεία με `id` το οποίο: (α) περιέχεται στη δομή `this`, αλλά όχι στη δομή `h`, ή (β) περιέχεται και στις δύο δομές: η προτεραιότητα του στοιχείου στην επιστρεφόμενη δομή ισούται με τη διαφορά των προτεραιοτήτων του στοιχείου στις `this` και `h`.

`public HashHeap kbest(int k)` Επιστρέφει νέο στιγμιότυπο κλάσης `HashHeap` που περιέχει τα  $k$  στοιχεία της δομής `this` με τις μεγαλύτερες τιμές προτεραιότητας. Εφόσον η `this` έχει λιγότερα από  $k$  στοιχεία, η επιστρεφόμενη δομή περιέχει όλα αυτά τα στοιχεία.

## Διαδικαστικά Θέματα

**ΠΑΡΑΔΟΤΕΟ.** Θα παραδώσετε αρχείο `AM_Επώνυμο_Όνομα.zip` (όπου AM ο αριθμός μητρώου) που θα περιλαμβάνει ένα (1) αρχείο μόνο: Το αρχείο πηγαίου κώδικα `HashHeap.java`, επαρκώς σχολιασμένο.

**ΠΡΟΣΟΧΗ** στο παραδοτέο σας:

- Μην χρησιμοποιήσετε άλλη μορφή συμπίεσης εκτός του `.zip`.
- Μην παραδώσετε στο `.zip` ολόκληρο το `project folder` στο οποίο εργάζεστε σε IDE της επιλογής σας: ζητείται μόνο το αρχείο `HashHeap.java`.

**ΠΑΡΑΔΟΣΗ** αποκλειστικά μέσω της πλατφόρμας «ΑΡΙΣΤΑΡΧΟΣ» του τμήματος, έως και την **8/1/2025, 23:59**. Ανεβάστε το `AM_Επώνυμο_Όνομα.zip` στην περιοχή «Εργασίες».

**ΕΡΩΤΗΣΕΙΣ/ΑΠΟΡΙΕΣ/ΔΙΕΥΚΡΙΝΙΣΕΙΣ** αποκλειστικά μέσα από την Περιοχή Συζήτησης «Εργασία 2 (2024-2025)» της πλατφόρμας «ΑΡΙΣΤΑΡΧΟΣ». Δε θα απαντηθούν emails με απορίες.

**ΔΕ ΘΑ ΑΞΙΟΛΟΓΗΘΟΥΝ ΕΡΓΑΣΙΕΣ ΠΟΥ:** (i) παραδίδονται εκπρόθεσμα ή με άλλο τρόπο (π.χ. email), (ii) παραδίδονται σε μορφή συμπίεσης διαφορετική από `.zip`, (iii) περιλαμβάνουν άσχετα αρχεία.

**ΒΑΘΜΟΛΟΓΗΣΗ** Η εργασία συνεισφέρει 20% στον τελικό βαθμό (εφόσον το γραπτό σας βαθμολογηθεί με τουλάχιστον 4). Η εργασία βαθμολογείται με 0, αν δεν δύναται να αξιολογηθεί, π.χ., αν ο κώδικας δεν μεταγλωττίζεται ορθά.

**Η εργασία είναι αυστηρά ατομική.** Αντιγραφή επιφέρει άμεσο μηδενισμό και των δύο εργασιών του μαθήματος. Τα παραδοτέα των εργασιών θα ελεγχθούν (όλα ανά ζεύγη).