

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΔΕΥΤΕΡΗ ΕΡΓΑΣΙΑ (2024-25)

Η δεύτερη εργασία αποτελεί συνέχεια της πρώτης όπου τώρα καλείσθε να οργανώσετε την πληροφορία των παικτών χρησιμοποιώντας μία δομή η οποία λειτουργεί ταυτόχρονα ως δέντρο ελαχίστων ως προς την παγκόσμια κατάταξη των χρηστών και ως δυαδικό δέντρο αναζήτησης ως προς τις χρονικές σημάνσεις εγγραφής. Συγκεκριμένα σε κάθε κόμβο i του δέντρου αποθηκεύονται όλες οι πληροφορίες ενός χρήστη και έστω r_i , t_i είναι η παγκόσμια κατάταξη και η χρονική σήμανση εγγραφής του χρήστη. Ως προς τις κατατάξεις r_i , η δομή δεδομένων λειτουργεί ως δέντρο ελαχίστων ενώ ως προς τις τιμές t_i , σχηματίζει ένα δένδρο δυαδικής αναζήτησης.

Στο σχήματα που έχουν συμπεριληφθεί στο τέλος της εργασίας, φαίνονται τα βήματα για τη διαγραφή του χρήστη i με $r_i=53$ και $t_i=13$. Ουσιαστικά ο κόμβος υποβιβάζεται διαδοχικά στο δέντρο με μία σειρά διαδοχικών απλών περιστροφών μέχρι να γίνει φύλλο του δέντρου όποτε και εύκολα διαγράφεται. Σε κάθε βήμα, η περιστροφή αφορά τον κόμβο του ζεύγους (53,13) και ένα από τα δύο παιδιά του, συγκεκριμένα όποιο έχει την υψηλότερη κατάταξη. Π.χ. στο βήμα (α) ο κόμβος (53,13) έχει δύο παιδιά το (58,10) και το (64,24). Η περιστροφή γίνεται μεταξύ του κόμβου (53,13) και (58,10) δηλ. με το παιδί που έχει την υψηλότερη κατάταξη.

Στο ίδιο σχήμα φαίνεται και η διαδικασία της εισαγωγής ακολουθώντας τα βήματα από το τέλος προς την αρχή. Συγκεκριμένα στο δέντρο του σχήματος (στ), γίνεται εισαγωγή του χρήστη i με $r_i=47$ και $t_i=13$, λαμβάνονται υπόψη μόνο τις τιμές t_i . Στη συνέχεια, με μία σειρά περιστροφών ο νέος παίκτης ανεβαίνει ψηλότερα στο δέντρο μέχρι του ψηλότερου σημείου στο οποίο δεν παραβιάζεται η βασική ιδιότητα του δέντρου ελαχίστων (βήματα (στ)-(α)).

Πρέπει να σημειωθεί ότι είναι πιθανόν δύο ή περισσότεροι χρήστες να έχουν την ίδια χρονική σήμανση εγγραφής και επομένως όλες οι λειτουργίες δυαδικού δέντρου αναζήτησης πρέπει να επεκταθούν ώστε να καλύπτουν την περίπτωση πολλαπλών αντιγράφων της ίδιας τιμής στο δέντρο. Αντίθετα, θεωρείστε ότι παγκόσμια κατάταξη είναι μοναδική για κάθε χρήστη¹. Επίσης μπορεί να αποδειχθεί ότι αν οι κατατάξεις των χρηστών είναι τυχαίες, η δομή δεδομένων είναι ένα ζυγισμένο δυαδικό δέντρο.

Θα πρέπει να υλοποιηθούν οι ακόλουθες λειτουργίες:

α) Εισαγωγή νέου παίκτη. Θεωρείστε ότι η δομή δεδομένων είναι αρχικά άδεια και οι νέοι παίκτες θα εισαχθούν στο δέντρο με τη σειρά με την οποία είναι αποθηκευμένοι στο αρχείο `contest_participants.txt`.

β) Διαγραφή ενός παίκτη. Συγκεκριμένα η λειτουργία θα δέχεται ως είσοδο ένα δείκτη ως προς το κόμβο του δέντρου που πρόκειται να διαγραφεί και στη συνέχεια ο κόμβος και οι

¹ Το πρόγραμμα `CreateContestantsInput.py` έχει αλλάξει ώστε να παράγει τυχαίες μοναδικές κατατάξεις στους χρήστες.

πληροφορίες του αντίστοιχου χρήστη θα διαγράφονται από το δέντρο με τη διαδικασία που περιεγράφηκε παραπάνω.

γ) Επιστροφή του ID του παίκτη που έχει την k -οστή καλύτερη κατάταξη μεταξύ όλων των συμμετεχόντων στο τουρνουά αθλημάτων, όπου η θετική ακέραια παράμετρος k θα δίνεται ως είσοδος στη λειτουργία. Η υλοποίηση της συγκεκριμένης λειτουργίας θα πρέπει να χρησιμοποιεί $O(k)$ επιπλέον χώρο ενώ η πολυπλοκότητα χειρότερης περίπτωσης θα είναι $O(k \log_2 k)$. Αποδείξτε ότι η υλοποίησή σας επιτυγχάνει αυτές τις πολυπλοκότητες χώρου και χρόνου.

δ) Εκτύπωση των ID των χρηστών που η χρονική σήμανση εγγραφής τους είναι στο χρονικό διάστημα $[t_1, t_2]$ όπου οι παράμετροι t_1 και t_2 θα δίνονται ως είσοδος στη λειτουργία. Θα πρέπει η λειτουργία να ολοκληρώνεται σε χρόνο $O(\#output + h)$ όπου $\#output$ είναι το πλήθος των χρηστών που επιστρέφονται ως έξοδος της λειτουργίας και h το ύψος του δέντρου όταν εκτελέστηκε η λειτουργία. Επιτρέπεται να χρησιμοποιήσετε το πολύ $O(h)$ επιπλέον χώρο. Αποδείξτε ότι η υλοποίησή σας ικανοποιεί αυτούς τους περιορισμούς.

ε) Διαγραφή όλων των χρηστών που η εγγραφή τους έγινε μια συγκεκριμένη ημέρα η οποία θα δίνεται ως είσοδος στη λειτουργία. Στην υλοποίηση σας εκμεταλλευτείτε το γεγονός ότι υπάρχουν μαζικές διαγραφές στο δέντρο κατά την εκτέλεση της λειτουργίας ώστε να μειώσετε το χρόνο εκτέλεσης σε σχέση με μία υλοποίηση που διαγράφει επαναληπτικά με τη χρήση της λειτουργίας διαγραφής (ερώτημα β) ένα προς ένα τους παίκτες. Προσδιορίστε την πολυπλοκότητα χρόνου και επιπρόσθετου χώρου της μεθόδου σας. Αποδείξτε τον ισχυρισμό σας.

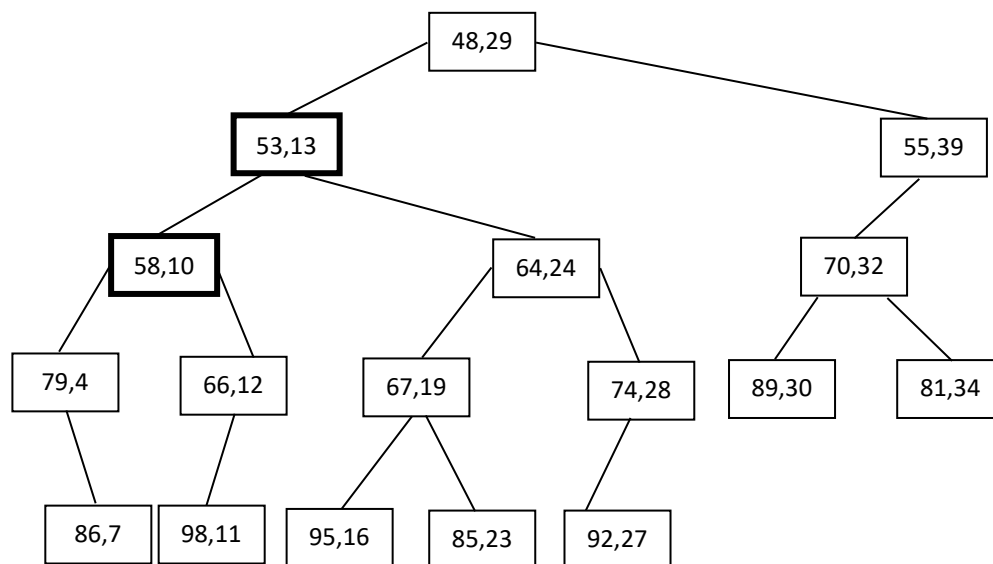
στ) Εκτύπωση του ID του χρήστη με την καλύτερη κατάταξη μεταξύ όλων των χρηστών που έχουν εγγραφεί στο σύστημα πριν μία συγκεκριμένη ημερομηνία η οποία θα δίνεται ως είσοδος στη λειτουργία. Προσδιορίστε την πολυπλοκότητα χρόνου και επιπρόσθετου χώρου της μεθόδου σας. Αποδείξτε τον ισχυρισμό σας.

ΠΑΡΑΔΟΤΕΑ

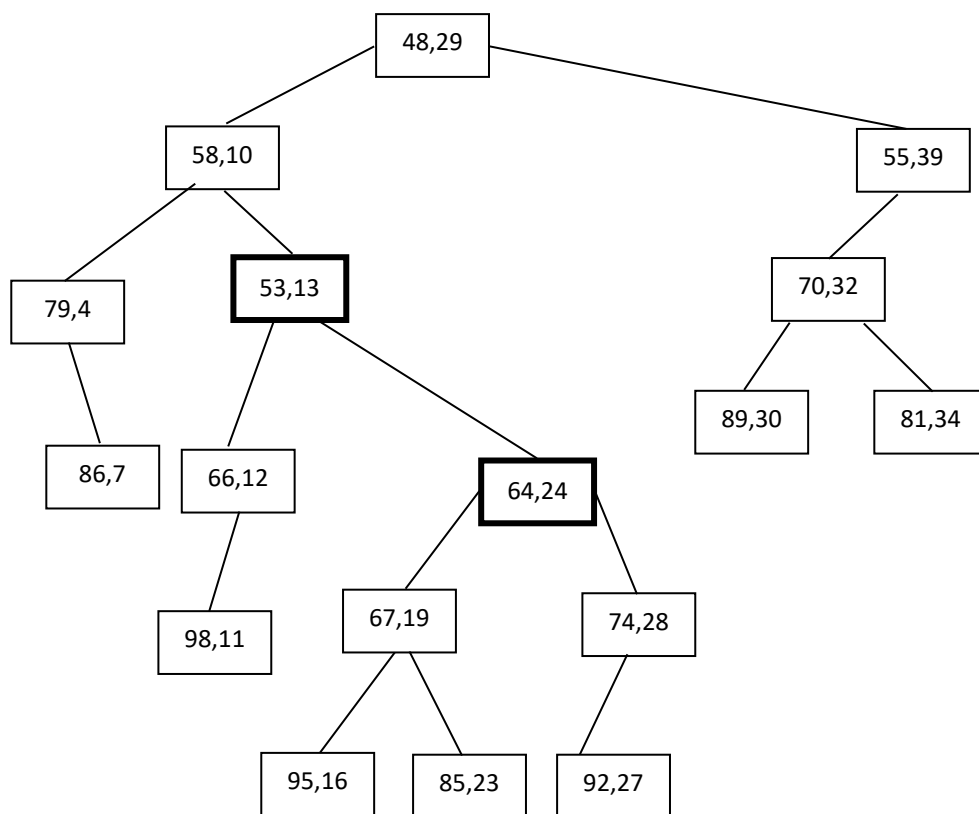
Θα πρέπει να παραδοθεί ο πηγαίος κώδικας του προγράμματος σας. Ιδιαίτερη βαρύτητα θα πρέπει να δοθεί στη σωστή τεκμηρίωση του προγράμματος σας. Εντός του πηγαίου κώδικα θα πρέπει να συμπεριληφθούν «πυκνά» σχόλια ουσίας. Επιπλέον, ο κώδικας σας θα πρέπει να συνοδεύεται από ξεχωριστό κείμενο που θα παρέχει λεπτομερή περιγραφή των τεχνικών σας. Συμπεριλάβετε τα παραπάνω αρχεία σε ένα συμπιεσμένο αρχείο, το οποίο θα είναι το συνολικό παραδοτέο σας. Μην συμπεριλάβετε το εκτελέσιμο αρχείο του προγράμματος σας. Η παράδοση των εργασιών θα γίνει μέσω του Thales.

Η εργασία μπορεί να εκπονηθεί από ομάδα μέχρι δύο ατόμων αυστηρώς. Συμπεριλάβετε τα στοιχεία όλων των μελών της ομάδας (Ονοματεπώνυμο και ΑΜ) στο εξώφυλλο του κείμενου που θα παραδώσετε.

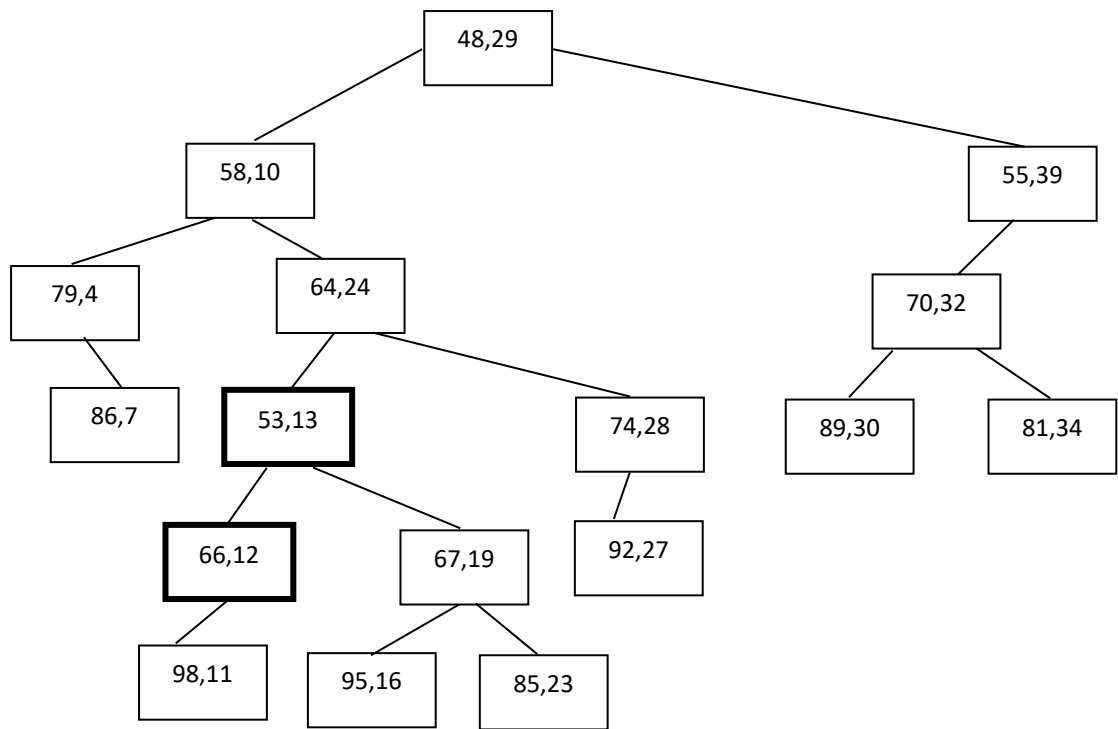
Προθεσμία Παράδοσης: Σάββατο 31 Μαΐου 2025



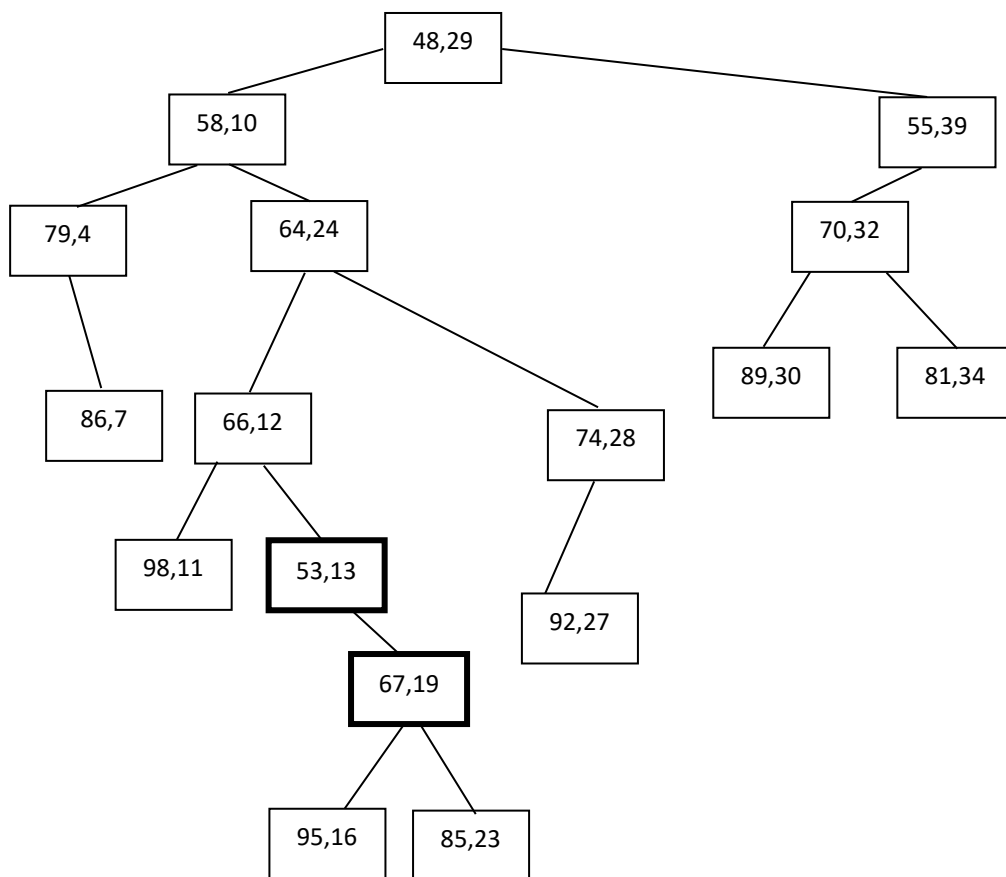
(α)



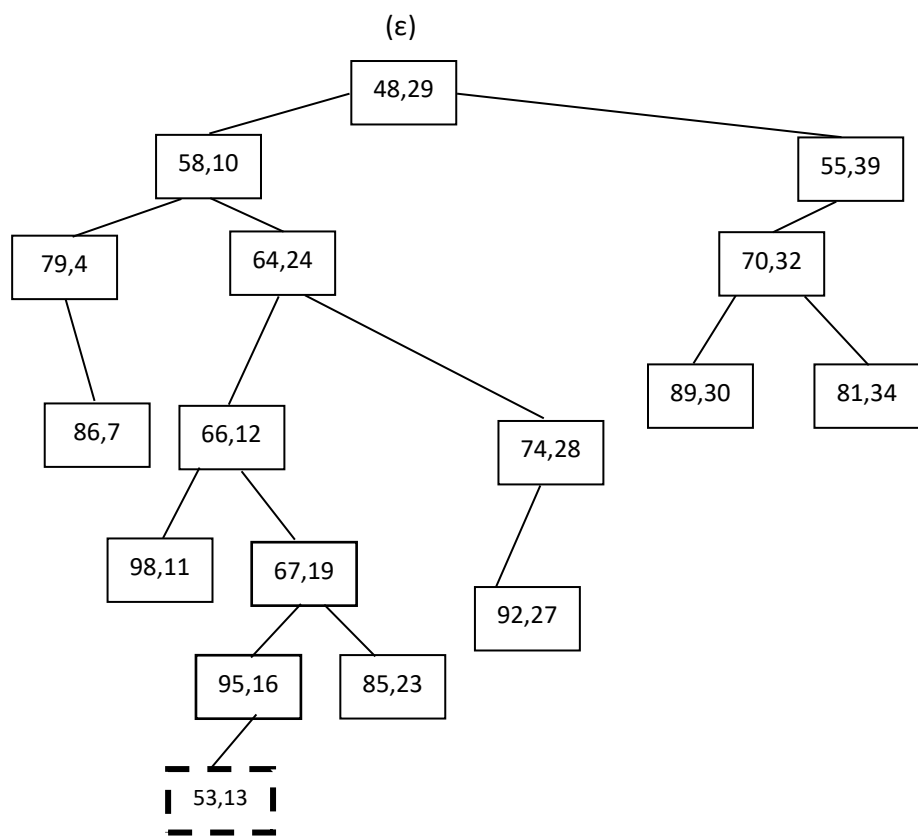
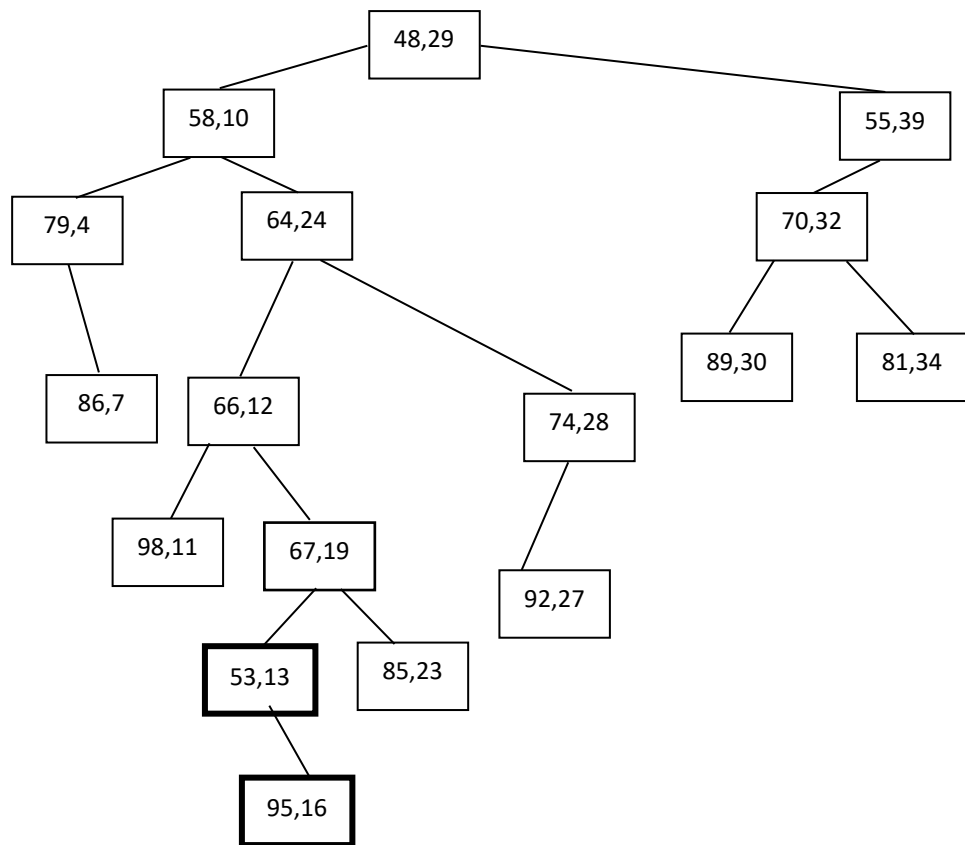
(β)



(γ)



(δ)



(σ)