

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
ΔΕΥΤΕΡΗ ΕΡΓΑΣΙΑ
(2021-22)

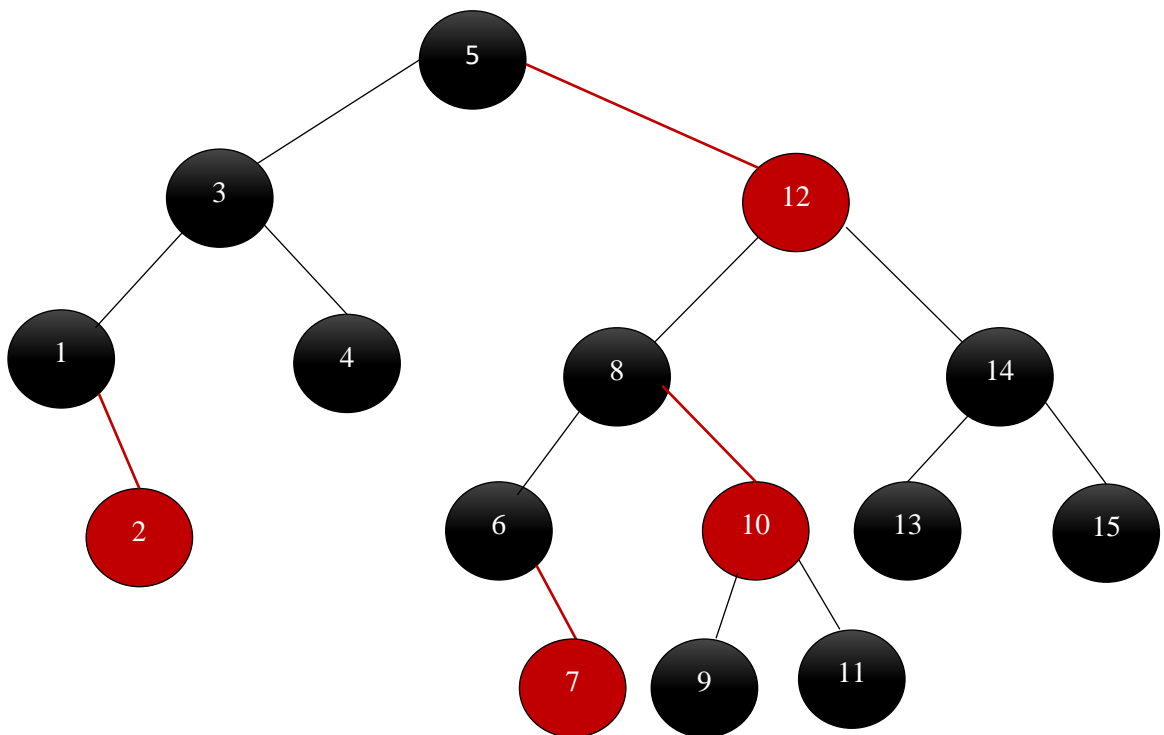
A. Υλοποίηση των AA-trees

Στο πρώτο μέρος της εργασίας, καλείστε να υλοποιήσετε τις βασικές λειτουργίες (αναζήτηση, εισαγωγή, διαγραφή) της δομής δεδομένων AA-Tree.

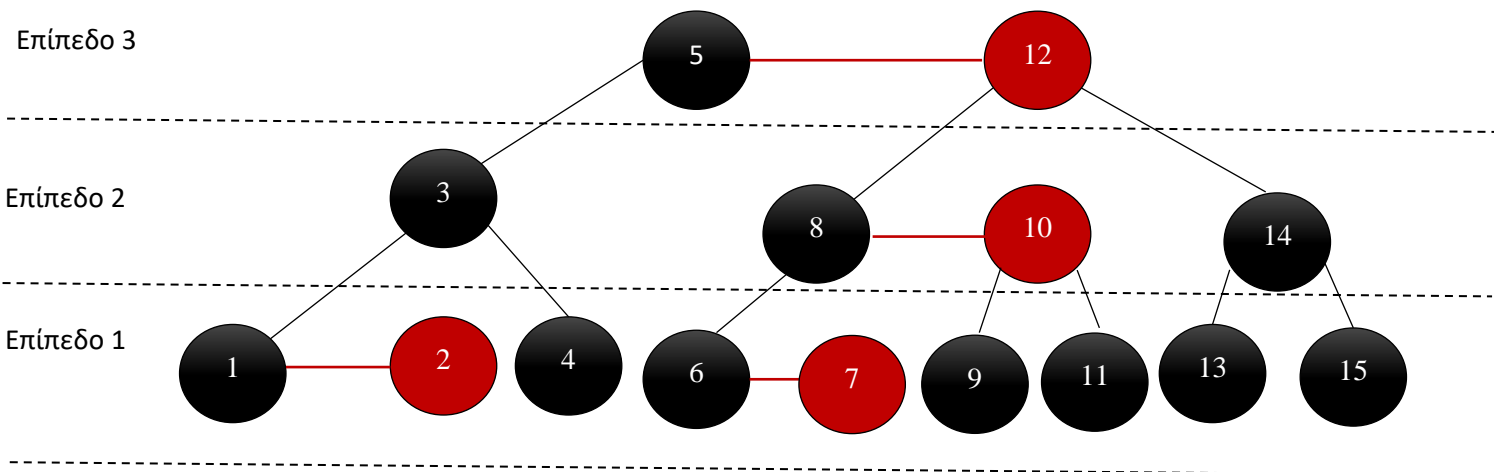
Τα δέντρα AA-Trees προτάθηκαν ως απλοποίηση των Red-Black δέντρων, μειώνοντας το πλήθος των περιπτώσεων που πρέπει να ληφθούν υπόψη όταν απαιτείται επαναζύγιση. Ισχύουν οι ίδιες ιδιότητες με τα RB δέντρα με επιπλέον μία νέα ιδιότητα. Συγκεκριμένα:

- 1) Κάθε κόμβος χρωματίζεται είτε κόκκινος είτε μαύρος.
- 2) Η ρίζα είναι μαύρη.
- 3) Οι εξωτερικοί κόμβοι είναι μαύροι.
- 4) Αν ο κόμβος είναι κόκκινος, τα παιδιά του πρέπει να είναι μαύρα.
- 5) Το πλήθος των μαύρων κόμβων στο μονοπάτι από οποιοδήποτε κόμβο σε οποιοδήποτε εξωτερικό κόμβο (απόγονο του πρώτου κόμβου) είναι το ίδιο.
- 6) Τα αριστερό παιδί ενός κόμβου δεν μπορεί να είναι κόκκινο.

Ένα παράδειγμα δέντρου AA-tree είναι:



Στην πραγματικότητα δεν απαιτείται η πληροφορία του χρώματος και αρκεί να αποθηκεύεται η πληροφορία του επιπέδου κάθε κόμβου με τους κόκκινους κόμβους να είναι στο ίδιο επίπεδο με τους γονείς τους, δηλ.:



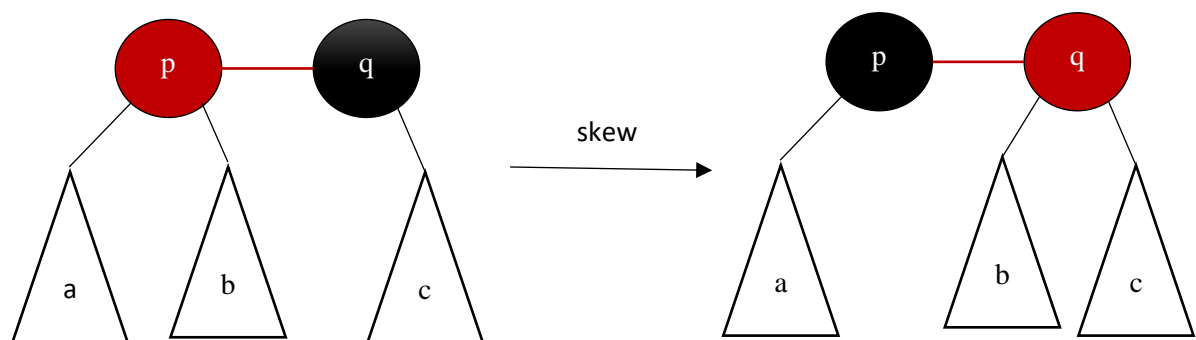
Μπορούν να γίνουν οι εξής παρατηρήσεις:

1. Οι οριζόντιοι δείκτες είναι πάντα οι δείκτες προς τα δεξιά παιδιά.
2. Δεν υπάρχουν διαδοχικοί οριζόντιοι δείκτες.
3. Κόμβοι σε επίπεδα υψηλότερα του επιπέδου 1 πρέπει να έχουν δύο παιδιά.
4. Αν ένας κόμβος δεν έχει δεξιό οριζόντιο δείκτη, τα δύο παιδιά του είναι στο ίδιο επίπεδο.

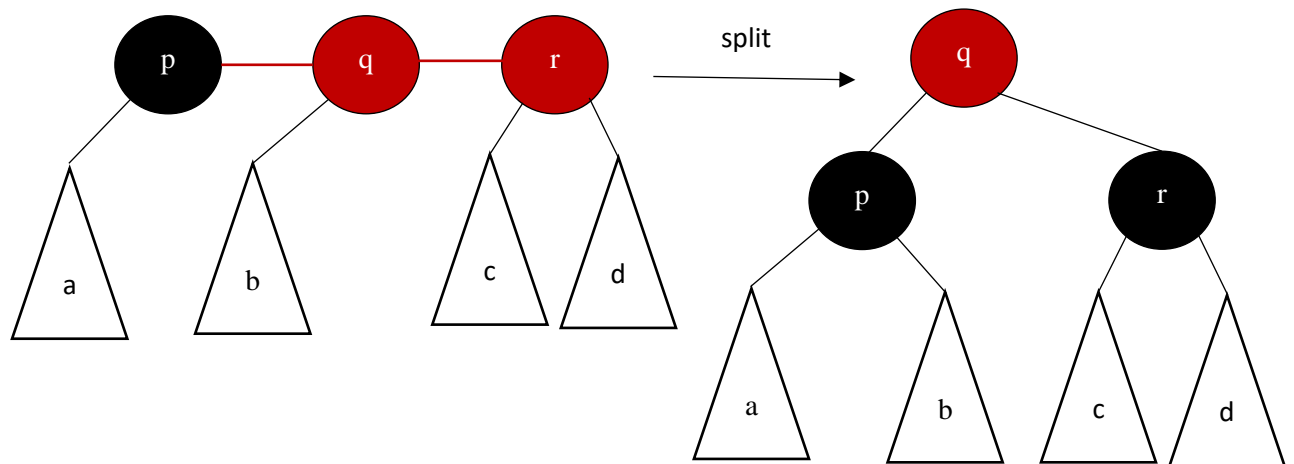
Εισαγωγή ενός στοιχείου

Το νέο στοιχείο εισάγεται πάντα ως κόκκινος κόμβος και επομένως ενδέχεται να δημιουργηθεί είτε αριστερός οριζόντιος δείκτης ή συνεχόμενοι δεξιοί οριζόντιοι δείκτες. Και στις δύο περιπτώσεις παραβιάζονται οι βασικές ιδιότητες του AA-tree. Η διόρθωση αρχίζει από το σημείο εισαγωγής και ακολουθώντας το μονοπάτι προς τη ρίζα του δέντρου, ενδέχεται να χρειαστεί να διανύσει όλο το μονοπάτι και να ολοκληρωθεί με την επίσκεψη της ρίζας. Σε κάθε επίπεδο που υπάρχει πρόβλημα, εκτελούνται οι ενέργειες skew και split με αυτή τη σειρά.

Η λειτουργία skew διορθώνει το πρόβλημα των οριζόντιων αριστερών δεικτών. Η διορθωτική ενέργεια φαίνεται στο ακόλουθο σχήμα:



Η ενέργεια split διορθώνει το πρόβλημα των διαδοχικών δεξιών δεικτών. Η ενέργεια αυτή έχει ως εξής:



Διαγραφή στοιχείου

Η διαγραφή ενός στοιχείου ανάγεται ως γνωστό στη διαγραφή ενός κόμβου βαθμού ένα ή σε διαγραφή φύλλου. Στην πρώτη περίπτωση, ο κόμβος είναι στο επίπεδο 1 και διαθέτει ένα δεξιό κόκκινο παιδί. Το πρόβλημα λύνεται με επαναχρωματισμό του παιδιού. Στη δεύτερη περίπτωση, αν ο προς διαγραφή κόμβος είναι κόκκινος τότε η διαγραφή εκτελείται χωρίς πρόβλημα. Αν το φύλλο είναι ένας μαύρος κόμβος, απαιτούνται επανορθωτικές ενέργειες με κατεύθυνση προς τη ρίζα. Συγκεκριμένα αν ένας κόμβος που είναι σε επίπεδο υψηλότερο του χαμηλότερου (επιπέδου 1) πάψει να έχει δύο παιδιά μετά τη διαγραφή, θα πρέπει να κατέβει ένα επίπεδο ώστε να εκλείψει το πρόβλημα αυτό. Αυτή ενέργεια αυτή μπορεί να δημιουργήσει αριστερούς οριζόντιους δείκτες ή/και συνεχόμενους δεξιούς δείκτες. Σε κάθε επίπεδο, εφόσον προκύψουν αυτά τα προβλήματα απαιτείται αρχικά η εκτέλεση τριών λειτουργιών skew και στη συνέχεια δύο λειτουργιών split.

Για περισσότερες λεπτομέρειες σχετικά με το δέντρο AA-tree, μπορείτε να δείτε την αναφορά [1] η οποία είναι η εργασία στην οποία παρουσιάστηκε για πρώτη φορά το δέντρο αυτό. Την εργασία αυτή θα τη βρείτε αναρτημένη στο gunet μαζί με την εκφώνηση της εργασίας.

Β. Υλοποίηση ενός τρισδιάστατου εικονικού κόσμου

Θεωρείστε ένα παιχνίδι όπου ο ήρωας κινείται σε ένα τρισδιάστατο κόσμο ο οποίος αποδίδεται από ένα κύβο διάστασης n . Ο κόμβος υποδιαιρείται σε κελιά διαστάσεων $1 \times 1 \times 1$ και ο ήρωας ανά πάσα χρονική στιγμή είναι σε ένα από τα κελιά αυτά. Υποτίθεται επίσης κάθε κελί περιέχει λεία η αξία της οποίας αποτιμάται με κάποιους πόντους. Ο ήρωας καρπώνεται όλη τη λεία του κελιού στο οποίο βρίσκεται με αποτέλεσμα να αυξάνονται αντιστοίχως οι πόντοι που έχει συγκεντρώσει μέχρι εκείνη τη στιγμή. Στην περίπτωση εύρεσης λείας στο κελί, νέα λεία αξίας τυχαίων πόντων δημιουργείται και τοποθετείται με τυχαίο επίσης τρόπο σε ένα από τα κελιά του κύβου και σωρεύεται στην προηγούμενη λεία που τυχόν υπάρχει στο επιλεγμένο κελί. Αμέσως μετά ο ήρωας κινείται με τυχαίο τρόπο σε ένα από τα 26 γειτονικά κελιά (ή λιγότερα όταν είναι στις άκρες του κύβου) του κελιού στο οποίο βρίσκεται εκείνη τη στιγμή.

Υποθέστε ότι αρχικά υπάρχουν n μη κενά κελιά. Υλοποιήστε το παραπάνω σενάριο, χρησιμοποιώντας $O(n)$ χώρο και χρόνο $O(\log n)$ για κάθε βασική ενέργεια που απαιτείται. Μια

πρόσθετη δυνατότητα που θα πρέπει να προσφέρεται είναι και εξής. Αν ο ήρωας είναι στο κελί (x,y,z) , θα επιστρέφεται το πλησιέστερο κελί στην κατεύθυνση του άξονα X που περιέχει λεία. Η λειτουργία αυτή θα πρέπει να έχει χρόνο εκτέλεσης $O(\log n)$.

Για την επίτευξη των παραπάνω επιδόσεων, θα χρησιμοποιήσετε τη δομή των AA-trees που υλοποιήσατε στο πρώτο τμήμα της εργασίας.

Παραδοτέα

Θα πρέπει να παραδοθεί ο πηγαίος κώδικας μαζί με τον εκτελέσιμο. Ιδιαίτερη βαρύτητα θα πρέπει να δοθεί στη σωστή τεκμηρίωση των προγραμμάτων σας. Θα πρέπει λοιπόν ο κώδικας σας να συνοδεύεται από ξεχωριστό κείμενο που θα παρέχει λεπτομερή περιγραφή των τεχνικών σας. Επίσης, εντός του πηγαίου κώδικα θα πρέπει να συμπεριληφθούν «πυκνά» σχόλια ουσίας. Η παράδοση των εργασιών θα γίνει μέσω του gunet. Η εργασία μπορεί να εκπονηθεί από ομάδα μέχρι **δύο ατόμων αυστηρώς**.

Προθεσμία παράδοσης: Παρασκευή, 15 Ιουλίου 2022.

Βιβλιογραφία

[1] Andersson, Arne. "Balanced search trees made simple." Workshop on Algorithms and Data Structures. Springer, Berlin, Heidelberg, 1993.