2η Άσκηση Υλοποίηση Συστημάτων Βάσεων Δεδομένων Χειμερινό 2015

Παπατσώρης Ιωάννης 1115201300137
Βραχάς Χρίστος 1115201300024

Μεταγλώττιση: gcc mainX.c AM.c util.c BF_Y.a
όπου $X \in [1, 4]$ και $Y \in \{32, 64\}$

Οργάνωση ειδικής πληροφορίας σε κάθε block:

Info block:

- **F_TYPE**: Αναγνωριστικό του αρχείου. Χαρακτήρας 'b' εάν πρόκειται για ευρετήριο βασισμένο σε B+ δένδρο.
- **TYPE_1**: Τύπος πρώτου πεδίου. Χαρακτήρας 'c', 'i' ή 'f' για string, integer ή float.
- **LEN_1**: Μήκος πρώτου πεδίου. Ακέραιος, 4 για integer ή float, 1-255 για string.
- **TYPE_2**: Τύπος δεύτερου πεδίου.
- **LEN_2**: Μήκος δεύτερου πεδίου.
- **ROOT**: Αναγνωριστικό του block που αποτελεί ρίζα του B+ δένδρου.

Data block:

- **BD_TYPE**: Αναγνωριστικό του τύπου block. Χαρακτήρας 'd' εάν πρόκειται για data block.
- **BD_RIGHT**: Αναγνωριστικό του data block που βρίσκεται δεξιά του τρέχοντος data block.
- **BD_ADDR**: Επόμενη κενή θέση στο block.

Index block:

- **BI_TYPE**: Αναγνωριστικό του τύπου block. Χαρακτήρας 'i' εάν πρόκειται για index block.
- **BI_ADDR**: Επόμενη κενή θέση στο block.
- **BI_P0**: Αναγνωριστικό του block στο οποίο δείχνει ο αριστερότερος 'δείκτης'.

Το πρώτο block είναι πάντα το info block. Κατά την πρώτη εισαγωγή εγγραφής, δημιουργείται data block το οποίο αποτελεί ρίζα. Όταν γεμίσει, σπάσει σε δύο και δημιουργείται το πρώτο index block, το οποίο αποτελεί τη καινούρια ρίζα.

Κατά την εισαγωγή εγγραφής, αρχικά γίνεται αναζήτηση του data block στο οποίο πρέπει να γίνει η εισαγωγή. Στην οντότητα *Path* φυλάσσεται το μονοπάτι index block που ακολουθήσαμε για να φτάσουμε στο στόχο data block, ώστε σε περίπτωση σπασίματος ενός block, να γνωρίζουμε το αμέσως προηγούμενό του. Σε περίπτωση που η εγγραφή χωράει στο data block, μπαίνει σε κατάλληλη θέση ώστε να διατηρείται η ταξινόμηση. Αλλιώς το data block σπάει σε δύο, μοιράζουμε $\left\lfloor \frac{n+1}{2} \right\rfloor$ εγγραφές στο πρώτο εφόσον είναι εφικτό, και τις υπόλοιπες στο δεύτερο, όπου n ο αριθμός των εγγραφών που χωράνε στο block. Φροντίζουμε οι διπλότυπες εγγραφές να μοιραστούν στο ίδιο block, οπότε μπορεί να χρειαστεί να ξεπεράσουμε τις $\left\lfloor \frac{n+1}{2} \right\rfloor$ εγγραφές σε κάποιο block.

Αφού μοιραστούν οι εγγραφές, εισάγουμε το κλειδί της πρώτης εγγραφής του δεύτερου block στο index block του παραπάνω επιπέδου σύμφωνα με το μονοπάτι που ακολουθήσαμε. Εάν χωράει, μπαίνει σε κατάλληλη θέση ώστε να διατηρείται η ταξινόμηση. Αλλιώς το index block σπάει σε δύο, μοιράζουμε $\left\lfloor \frac{n}{2} \right\rfloor$ ζευγάρια κλειδί-δείκτη σε κάθε block και το μεσαίο κλειδί εισάγεται στο index block του ακόμα παραπάνω επιπέδου. Εάν δεν υπάρχει δημιουργείται νέα ρίζα, εάν δε χωράει σπάει το index block κοκ.

Η OpenIndexScan, ανοίγει μια σάρωση η οποία ικανοποιεί τους "περιορισμούς" που δίνονται. Αφού γίνει η απαραίτητη καταχώρηση στον πίνακα ανοιχτών σαρώσεων, εντοπίζει, ανάλογα με τον operator, το block στο οποίο βρίσκεται η πρώτη εγγραφή που ικανοποιεί τους περιορισμούς, καθώς και την ίδια την εγγραφή, και ενημερώνει την καταχώρηση στον πίνακα σαρώσεων.

Η FindNextEntry με τη σειρά της, παίρνει το offset της τελευταίας εγγραφής, καθώς και το τελευταίο block, που αντιστοιχούν στη σάρωση που εξετάζεται, κι επιστρέφει αναλόγως, την τιμή του δεύτερου πεδίου.