



ΗΥ-240 Δομές Δεδομένων

Προγραμματιστική Εργασία Μέρος Β

Αντώνης Παπαϊωάννου

Διαδικαστικά

- ❖ Παράδοση: Δευτέρα, 13 Μαΐου 2019, ώρα 23:59.
- ❖ Compile και run σε μηχανήματα της σχολής
 - ❖ *Μέρος της βαθμολογίας*
- ❖ Τρόπος Παράδοσης: turnin
 - ❖ <http://csd.uoc.gr/~hy240b/current/submit.php>
- ❖ Online:
 - ❖ Main (java, C)
 - ❖ Test files
- ❖ <http://csd.uoc.gr/~hy240b>

Overview

- ❖ Προσομοίωση αγοράς
- ❖ Καταστήματα
- ❖ Προϊόντα
- ❖ Καταναλωτές
- ❖ Λίστα αγορών

Βασικές Δομές Δεδομένων

Shop

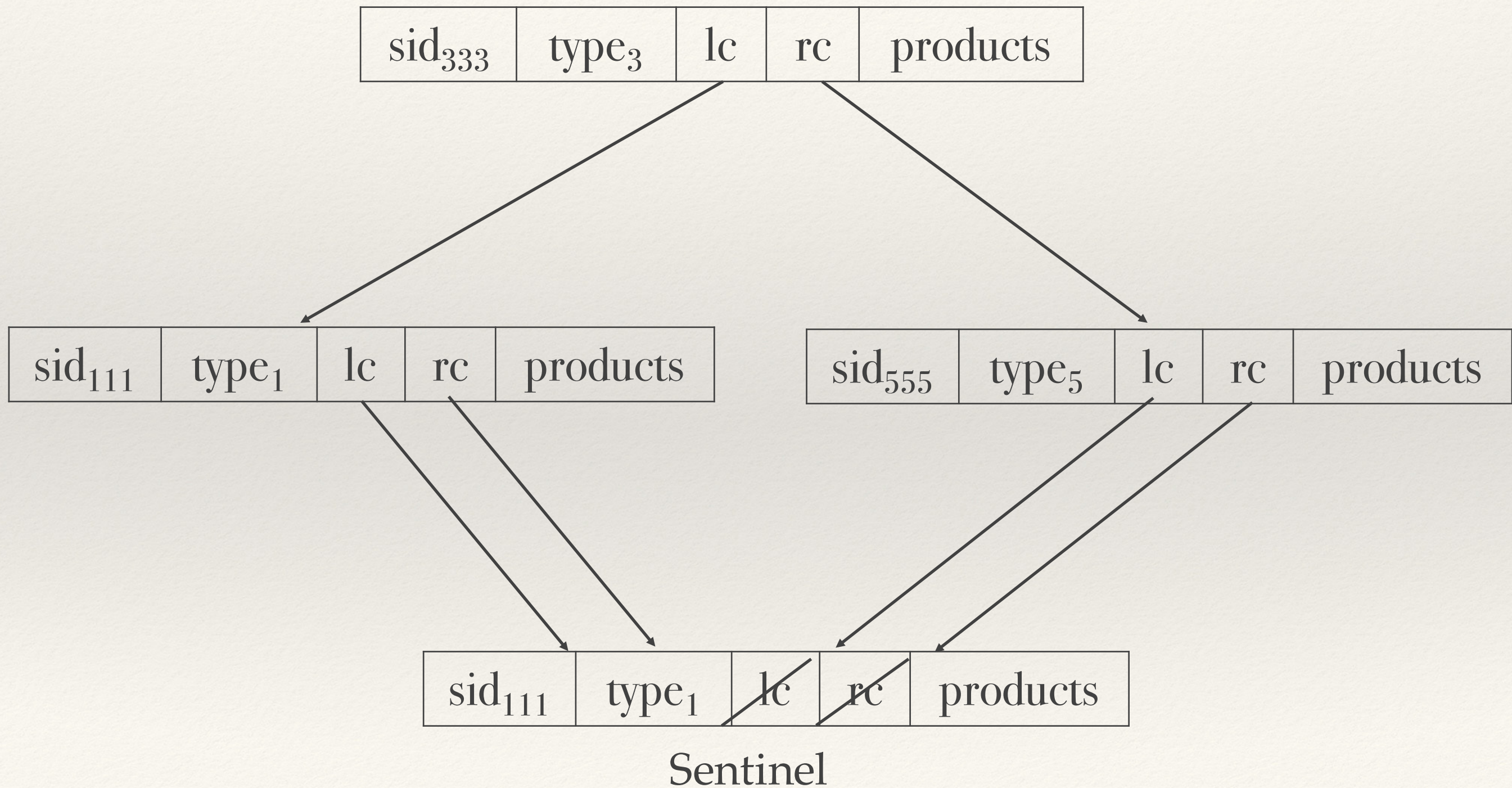
- ❖ sid: Μοναδικό αναγνωριστικό του καταστήματος (int)
- ❖ type: Ο τύπος του καταστήματος (ακέραιος, τιμές: 1-5)
- ❖ lc: : Δείκτης στο αριστερό παιδί του κόμβου (shop)
- ❖ rc: Δείκτης στο δεξί παιδί του κόμβου (shop)
- ❖ products: Λίστα προϊόντων. Απλά συνδεδεμένη, ταξινομημένη λίστα.
 - ❖ Όμοια με την πρώτη φάση
- ❖ Πρώτο ψηφίο του sid αντιστοιχεί στον τύπου του καταστήματος
 - ❖ Πχ. sid 512 αντιστοιχεί σε κατάστημα τύπου 5

sid ₃₃₃	type ₃	lc	rc	products
--------------------	-------------------	----	----	----------

Shops Tree

- ❖ Δυαδικό δέντρο αναζήτησης
- ❖ Κόμβος φρουρός
 - ❖ `sid = -1`
 - ❖ `lc, rc, products = null`

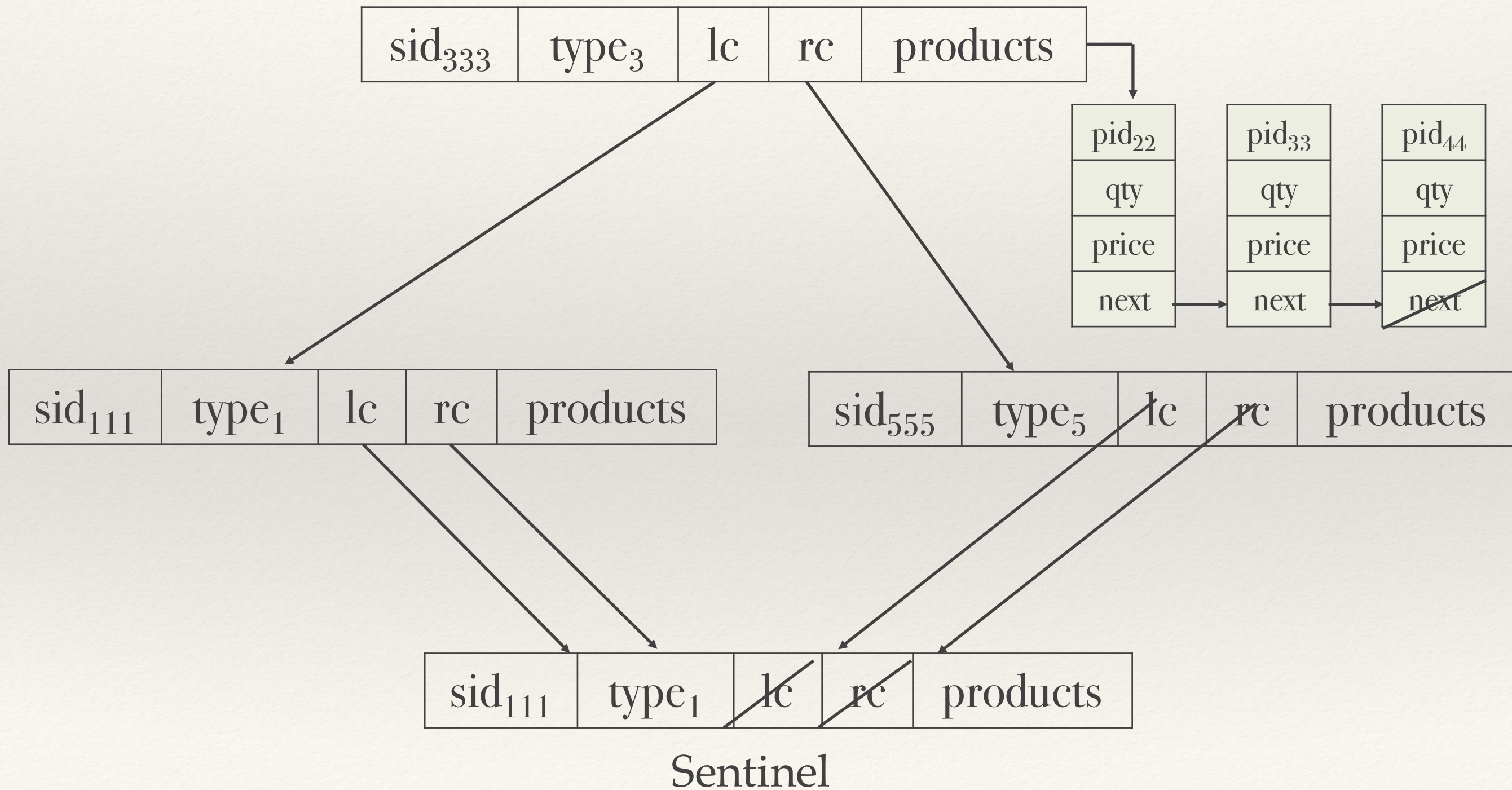
Shops Tree



Product

- ❖ pid: Μοναδικό αναγνωριστικό του προϊόντος (int)
- ❖ qty: Διαθέσιμη ποσότητα (int)
- ❖ price: Τιμή (int)
- ❖ next: Δείκτης στον επόμενο κόμβο της λίστας (product)

Shops Tree



Customers Hash Table

- ❖ Επιλογή μέγεθους
 - ❖ Διαθέσιμα: `max_customers_g`, `primes_g[160]`
- ❖ Επίλυση συγκρούσεων: Double Hashing
- ❖ Εγγραφές τύπου Consumer

Consumer

- ❖ cid: Μοναδικό αναγνωριστικό του καταναλωτή (int)
- ❖ shopping_tree: Δείκτης σε ρίζα δυαδικού δέντρου αναζήτησης (consumers tree)
- ❖ shopping_size: Μέγεθος του δέντρου (consumers tree)

cid ₂₂
shopping_size
shopping_tree

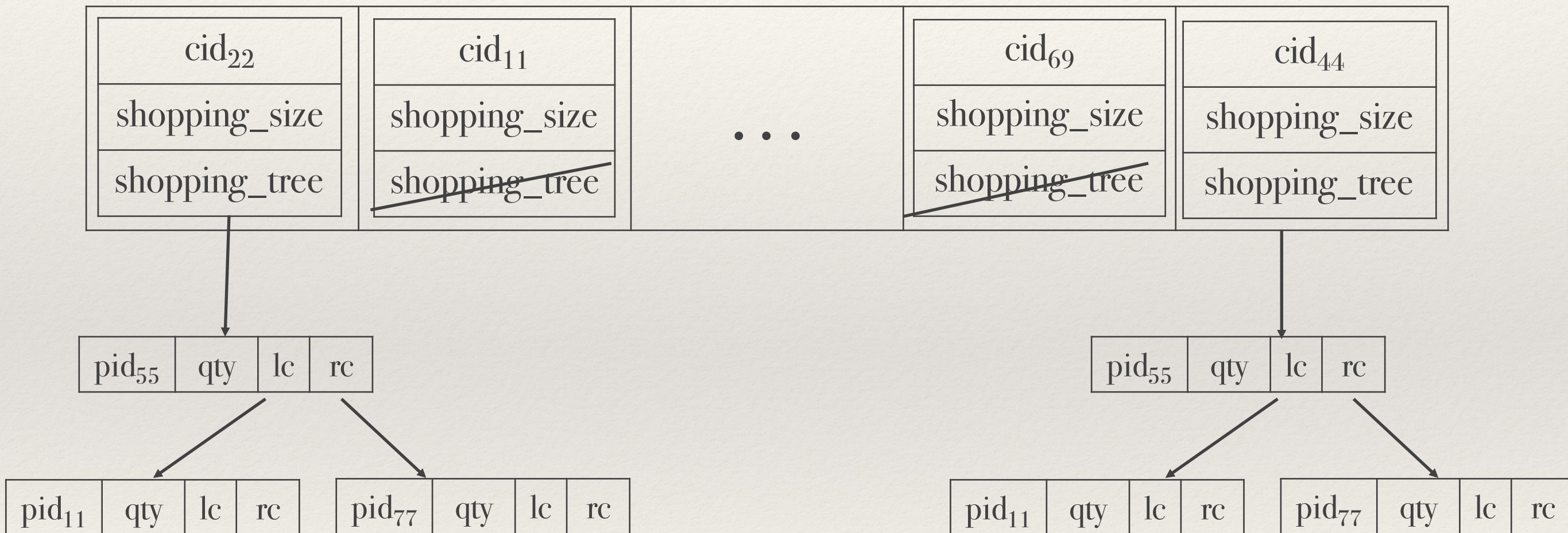
Customers Hashtable

cid ₂₂	cid ₁₁	...	cid ₆₉	cid ₄₄
shopping_size	shopping_size		shopping_size	shopping_size
shopping_tree	shopping_tree		shopping_tree	shopping_tree

Shopping Tree

- ❖ Δυαδικό δέντρο αναζήτησης
- ❖ pid: Μοναδικό αναγνωριστικό του προϊόντος (int)
- ❖ qty: Η ποσότητα που επιθυμεί ο καταναλωτής (int)
- ❖ lc: Δείκτης στο αριστερό παιδί του κόμβου
- ❖ rc: Δείκτης στο δεξί παιδί του κόμβου

Customers Hashtable



Γεγονότα

Γεγονότα: R, S

- ❖ R <sid>
 - ❖ Εισαγωγή νέου καταστήματος στο δέντρο καταστημάτων
- ❖ S <sid> <pid> <quantity> <price>
 - ❖ Εισαγωγή του προϊόντος pid στη λίστα προϊόντων του καταστήματος sid

Γεγονότα: C, L

- ❖ C <cid>
 - ❖ Εισαγωγή νέου καταναλωτή στο hashtable
- ❖ L <cid> <pid> <quantity>
 - ❖ Εισαγωγή του προϊόντος pid στο δέντρο προϊόντων του καταναλωτή cid.

Γεγονότα: D

- ❖ D <cid> <pid>
- ❖ Διαγραφή προϊόντος pid από το δέντρο του καταναλωτή cid.

Γεγονότα: G

- ❖ $G <cid>$
- ❖ Αναζήτηση του καταναλωτή cid στο hashtable
- ❖ Δημιουργία βοηθητικού πίνακα μεγέθους $shopping_size$
 - ❖ Αντιγραφή του δέντρου προϊόντων του καταναλωτή στον βοηθητικό πίνακα
 - ❖ Εισαγωγή σε ταξινομημένη διάταξη
 - ❖ Πολυπλοκότητα: $O(n)$
- ❖ Ενδοδιατεταγμένη διάσχιση στο δέντρο καταστημάτων
 - ❖ Για κάθε κόμβο του δέντρου

Γεγονότα: G

- ❖ Ενδοδιατεταγμένη διάσχιση στο δέντρο καταστημάτων
- ❖ Για κάθε κόμβο του δέντρου
 - ❖ Διατρέχετε τη λίστα προϊόντος του καταστήματος παράλληλα με το βοηθητικό πίνακα
 - ❖ Για τα κοινά προϊόντα ενημερώνεται τα πεδία qty
 - ❖ Πολυπλοκότητα: $O(m)$, m είναι το $\min(n, l)$, n = μέγεθος του βοηθητικού πίνακα, l = μέγεθος της λίστας του καταστήματος
- ❖ Για κάθε κόμβο του δέντρου που επισκέπτεστε, διατρέχετε τον πίνακα απ' την αρχή

Γεγονότα: M

- ❖ $M \langle cid1 \rangle \langle cid2 \rangle$
- ❖ Αναζήτηση των καταναλωτών $cid1$, $cid2$ στο hashtable
- ❖ Merge των δέντρων προϊόντων των καταναλωτών

Γεγονότα: M

- ❖ Δημιουργία βοηθητικού πίνακα μεγέθους ίσο με το μέγεθος του δέντρου προϊόντος του cid1 (χρήση του shopping_size)
 - ❖ cid1_array [shopping_size]
- ❖ Αντιγραφή στοιχείων του δέντρου στον πίνακα
 - ❖ Ταξινομημένη εισαγωγή
 - ❖ Πολυπλοκότητα: $O(n)$, n = μέγεθος του δέντρου
- ❖ Όμοια για τον καταναλωτή cid2 Δημιουργία βοηθητικού πίνακα μεγέθους shopping_size
 - ❖ cid2_array [shopping_size]

Γεγονότα: M

- ❖ Δημιουργία βοηθητικού πίνακα (merged_array) μεγέθους:
 - ❖ $\text{cid1_array} + \text{cid2_array}$
- ❖ Συνένωση των cid1_array και cid2_array στον merged_array
 - ❖ Ταξινομημένος
 - ❖ Πολυπλοκότητα: $O(n+m)$, $n = \text{μέγεθος cid1_array}$, $m = \text{μέγεθος cid2_array}$

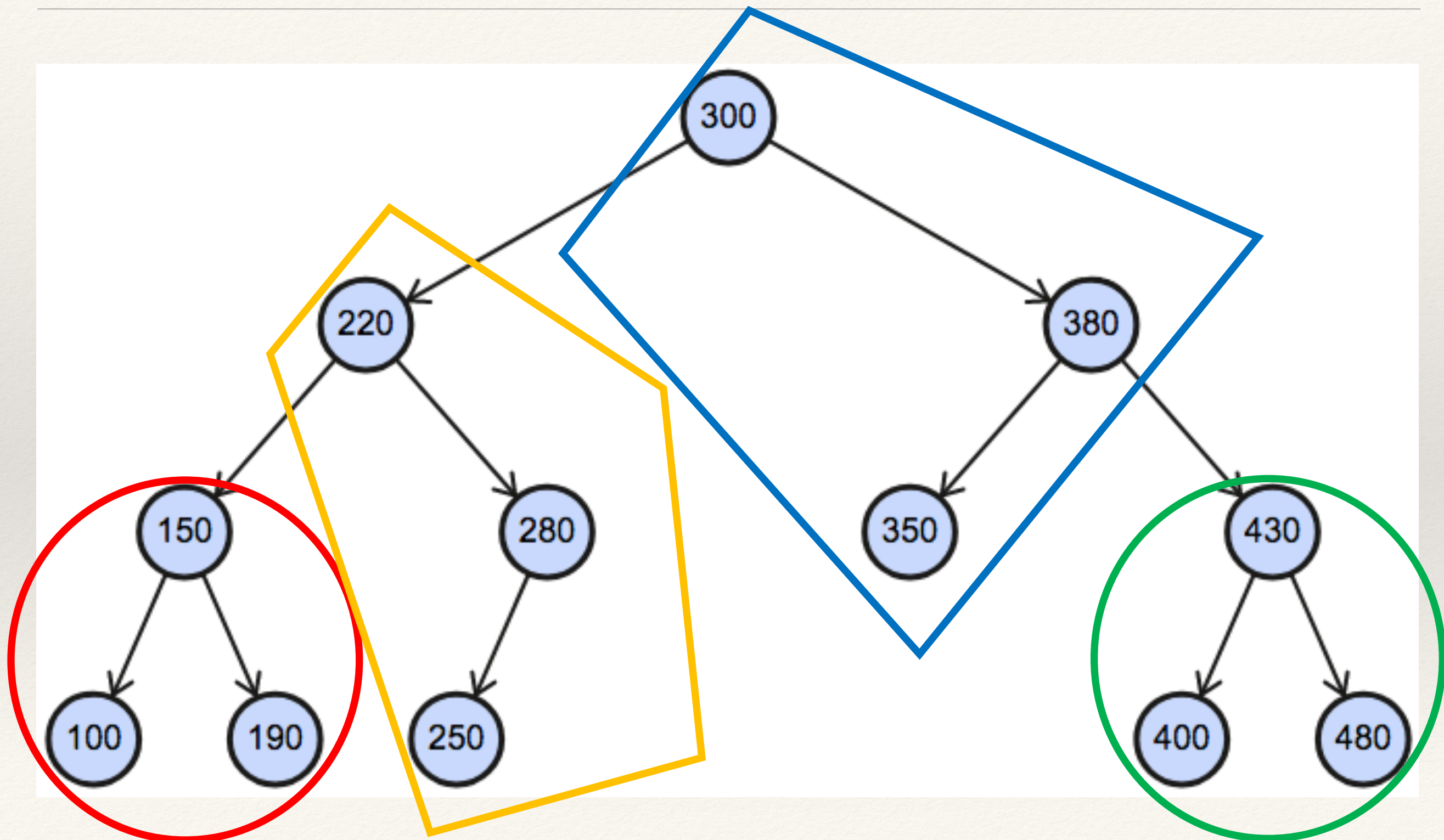
Γεγονότα: M

- ❖ Μετατροπή δυαδικού δέντρου αναζήτησης από το πίνακα `merged_array`
- ❖ Ελάχιστο ύψος
- ❖ Αντικατάσταση του δέντρου προϊόντων του `cid1` με το νέο δέντρο
- ❖ Διαγραφή του καταναλωτή `cid2`

Γεγονότα: T

- ❖ Bonus 15%
- ❖ Categorize stores
- ❖ Διαχωρισμός δέντρου καταστημάτων σε 5 μικρότερα δέντρα με βάση τον τύπο
- ❖ Πολυπλοκότητα: $O(h)$, h = ύψος του δέντρου
- ❖ Hint: Το 1ο ψηφίο του sid αντιστοιχεί στον τύπο του καταστήματος

Γεγονότα: T



Γεγονότα

❖ X

❖ Εκτύπωση όλων των καταστημάτων του δέντρου καταστημάτων

❖ Y

❖ Εκτύπωση των καταναλωτών από το hashtable

Διαδικαστικά

- ❖ Παράδοση: Δευτέρα, 13 Μαΐου 2019, ώρα 23:59.
- ❖ Compile και run σε μηχανήματα της σχολής
 - ❖ *Μέρος της βαθμολογίας*
- ❖ Τρόπος Παράδοσης: turnin
 - ❖ <http://csd.uoc.gr/~hy240b/current/submit.php>
- ❖ Online:
 - ❖ Main (java, C)
 - ❖ Test files
- ❖ <http://csd.uoc.gr/~hy240b>

