



Project Phase II

ΗΥ-240 – Δομές Δεδομένων

Διδάσκουσα: Παναγιώτα Φατούρου

Απόστολος Καράμπελας

apskar@csd.uoc.gr

10 Απριλίου 2020

Τμήμα Επιστήμης Υπολογιστών

Πανεπιστήμιο Κρήτης

1. Εισαγωγή
2. Περιγραφή Δομών
3. Δομές Αεροπορικών Εταιρειών
4. Γεγονότα
5. Είδη Γεγονότων

Εισαγωγή

- Ιστοσελίδα Μαθήματος: <http://www.csd.uoc.gr/~hy240b>
- Τρόπος Παράδοσης: Με το πρόγραμμα **turnin**
- Ημερομηνία Παράδοσης: Δευτέρα 18 Μαΐου 2020, ώρα 23:59
- Διαθέσιμες Υλοποιήσεις: C/C++ / JAVA
- Ερωτήσεις/Διευκρινήσεις: hy240b-list@csd.uoc.gr
- **ΠΡΟΣΟΧΗ:** Compile / Run στα μηχανήματα της σχολής

- Αεροπορικές εταιρείες που ανήκουν σε ένα όμιλο εταιρειών και πραγματοποιούν πτήσεις σε συγκεκριμένους προορισμούς.
- Η κάθε εταιρεία διαθέτει αεροπλάνα όπου κάθε αεροπλάνο εκτελεί πτήσεις προς ένα συγκεκριμένο προορισμό.
- Πελάτες της εταιρίας πραγματοποιούν πτήσεις προς τους διαθέσιμους προορισμούς και συγκεντρώνουν ταξιδιωτικά μίλια με κάθε ταξίδι.

Έχουμε λοιπόν τέσσερις βασικές οντότητες:

1. Αεροπορικές Εταιρείες – Airlines
2. Αεροπλάνα – Airplanes
3. Πτήσεις – Flights
4. Πελάτες – Clients

Περιγραφή Δομών

Οι γνώσεις που θα χρειαστούν για την υλοποίηση της 1ης φάσης της εργασίας, μεταξύ άλλων, είναι οι εξής:

- **Λίστες**

- Αναζήτηση, Εισαγωγή, Διαγραφή, Συγχώνευση
- Ταξινομημένες Λίστες
- Κόμβοι φρουροί

- **Δέντρα**

- Αναζήτηση, Εισαγωγή, Διαγραφή, Συγχώνευση
- Ταξινομημένα Δέντρα, Πλήρη Δέντρα, Νηματικά Δέντρα
- Δυαδικά Δέντρα, Τριαδικά Δέντρα

- **Ουρές Προτεραιότητας**

- Αναζήτηση, Εισαγωγή, Διαγραφή
- Ουρές Μέγιστης Προτεραιότητας
- Υλοποίηση Ουρών με Δέντρα και Πίνακες

Η εργασία αποτελείται από 3 ομάδες δομών:

1. Δομές που αφορούν τις αεροπορικές εταιρείες και τα αεροπλάνα
2. Δομές που αφορούν τις πτήσεις
3. Δομές που αφορούν τους πελάτες

Δομές Αεροπορικών Εταιρειών

Οι δομές αεροπορικών εταιρειών οργανώνονται ως εξής:

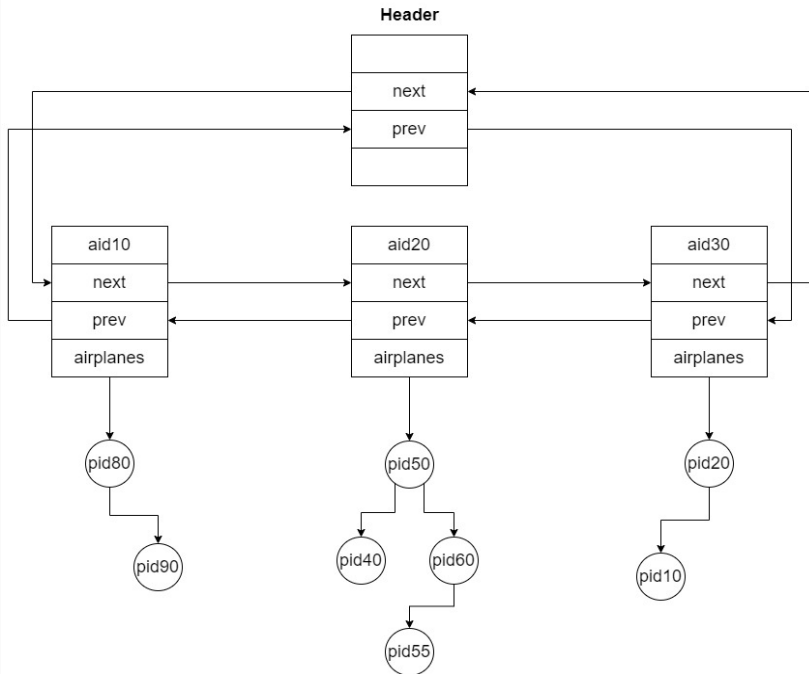
- Οι αεροπορικές εταιρείες αποθηκεύονται σε μια **κυκλική διπλά συνδεδεμένη ταξινομημένη λίστα**
- Η κάθε αεροπορική εταιρεία είναι ένα αντικείμενο τύπου **struct airline**
- Κάθε αεροπορική εταιρεία περιέχει ένα δείκτη που δείχνει στη ρίζα ενός **δυαδικού δέντρου αναζήτησης**, που ονομάζεται **δέντρο αεροπλάνων** (airplanes)
- Κάθε αεροπλάνο είναι ένα αντικείμενο τύπου **struct airplane**

Δομή Αεροπορικών Εταιρειών

aid	Αναγνωριστικό αεροπορικής εταιρείας (int)
airplanes	Δείκτης που δείχνει στη ρίζα ενός δυαδικού δένδρου αναζήτησης. (struct airplane*)
next	Δείκτης στον επόμενο κόμβο της λίστας (struct airline*)
prev	Δείκτης στον προηγούμενο κόμβο της λίστας (struct airline*)

Δομή Αεροπλάνων – Airplanes

- `pid` Αναγνωριστικό αεροπλάνου (int)
- `dest` Αναγνωριστικό προορισμού (int)
- `depart_time` Αναγνωριστικό που αντιστοιχεί στην ώρα αναχώρησης του αεροπλάνου. Η ώρα θα έχει τη μορφή HHMM. Για παράδειγμα, η ώρα 1620 αντιστοιχεί στην ώρα 16:20 (int)
- `lc` Δείκτης σε αριστερό θυγατρικό κόμβο (struct airplane*)
- `rc` Δείκτης σε δεξί θυγατρικό κόμβο (struct airplane*)



Ο Πίνακας Προορισμών:

- Έχει 10 θέσεις, μία για κάθε προορισμό.
- Σε κάθε θέση βρίσκεται ένας δείκτης στην **ρίζα ενός δέντρου πτήσεων** προς τον προορισμό που αντιστοιχεί στη θέση αυτή (struct flight*)

Τα Δέντρα Πτήσεων:

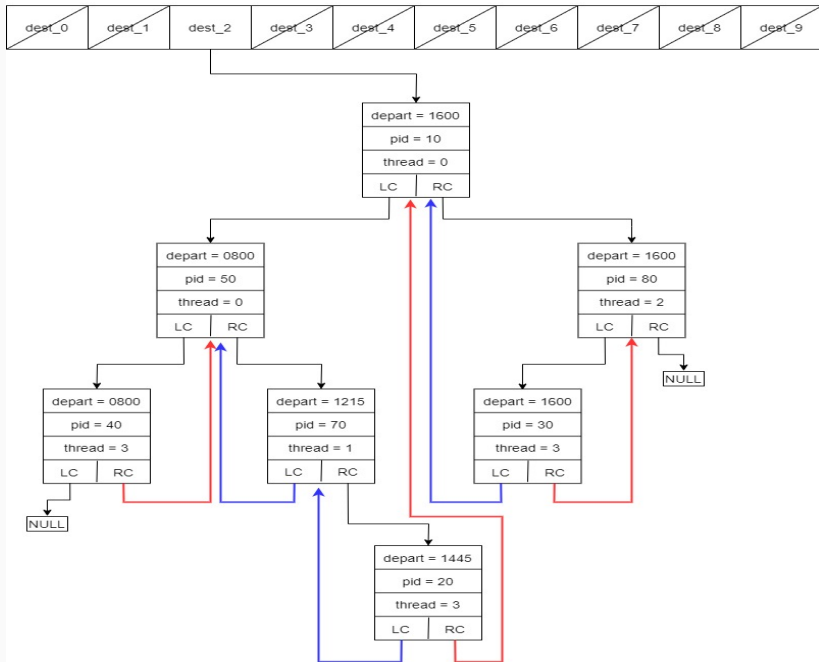
- Είναι ταξινομημένα δυαδικά νηματικά δέντρα
- Κάθε κόμβος του δέντρου πτήσεων είναι ένα αντικείμενο τύπου `struct flight`

- `pid` Αναγνωριστικό αεροπλάνου (int)
- `depart_time` Αναγνωριστικό που αντιστοιχεί στην ώρα αναχώρησης του αεροπλάνου. Η ώρα θα έχει τη μορφή HHMM. Για παράδειγμα, η ώρα 1620 αντιστοιχεί στην ώρα 16:20 (int)
- `lc` Δείκτης στον αριστερό θυγατρικό κόμβο (struct flight*)
- `rc` Δείκτης στο δεξί θυγατρικό κόμβο (struct flight*)

thread_status Πεδίο `int` με εύρος $[0, 3]$ που χαρακτηρίζει τους `lc` και `rc` δείκτες του κόμβου είτε ως νηματικούς είτε όχι. Οι νηματικοί δείκτες δείχνουν στον προηγούμενο ή επόμενο κόμβο αντίστοιχα, κατά την ενδοδιατεταγμένη διάσχιση

- Αν ο κόμβος έχει δύο κανονικά παιδιά, έχει `thread_status = 0`
- Αν ο κόμβος δεν έχει αριστερό παιδί, θα έχει `thread_status = 1` και ο `lc` δείκτης του είναι νηματικός.
- Αν ο κόμβος δεν έχει δεξί παιδί, θα έχει `thread_status = 2` και ο `rc` δείκτης του είναι νηματικός.
- Αν ο κόμβος δεν έχει κανένα παιδί, θα έχει `thread_status = 3` και οι `lc`, `rc` δείκτες του είναι νηματικοί.

DESTINATIONS

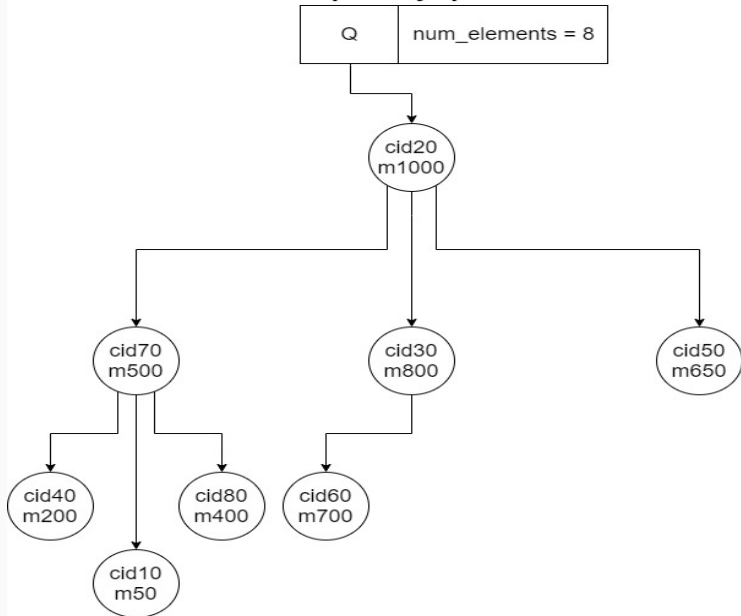


Η Δομή Πελατών:

- Είναι μία **Ουρά Μέγιστης Προτεραιότητας**
- Υλοποιείται με ένα **μερικώς διατεταγμένο πλήρες τριαδικό δέντρο**
- Κάθε κόμβος του δέντρου είναι ένα αντικείμενο τύπου **struct client**
- Η ταξινόμηση του δέντρου γίνεται **με βάση τα μίλια** που έχει συγκεντρώσει κάθε πελάτης, ώστε αυτός με τα περισσότερα μίλια να έχει τη μέγιστη προτεραιότητα (και να βρίσκεται στη ρίζα)
- Οι πελάτες συγκεντρώνουν $(i+1)*100$ μίλια με κάθε πτήση που εκτελούν σε προορισμό i

- cid** Αναγνωριστικό πελάτη (int)
- miles** Αναγνωριστικό που αναπαριστά τα μίλια που αντιστοιχούν στον πελάτη (int)
- lc** Δείκτης στον αριστερό θυγατρικό κόμβο (struct client*)
- mc** Δείκτης στο μεσαίο θυγατρικό κόμβο (struct client*)
- rc** Δείκτης στο δεξί θυγατρικό κόμβο (struct client*)

priority queue



Γεγονότα

Ένα γεγονός αποτελεί μία ενέργεια που εκτελείται μέσα στο σύστημά μας. Για παράδειγμα :

- Εισαγωγή μιας αεροπορικής εταιρείας
- Εισαγωγή ενός αεροπλάνου στον στόλο μιας εταιρείας
- κ.τ.λ.

- Τα γεγονότα αυτά είναι καταχωρημένα σε κάποια αρχεία (test files)
- Τα αρχεία υπάρχουν στη σελίδα του μαθήματος
- Το πρόγραμμα κάνει parse αυτά τα αρχεία και εκτελεί τη συνάρτηση που αντιστοιχεί στο εκάστοτε γεγονός
- Δεν χρειάζεται να γραφτεί επιπλέον κώδικας για το χειρισμό των γεγονότων
- Το μόνο που απουσιάζει είναι η υλοποίηση των συναρτήσεων

Ένα γεγονός έχει την εξής μορφή:

```
event_id [arithmetic_arguments [...]]
```

Είδη Γεγονότων

Εισαγωγή μιας αεροπορικής εταιρείας στον όμιλο

Register Airline

R aid

Ορίσματα:

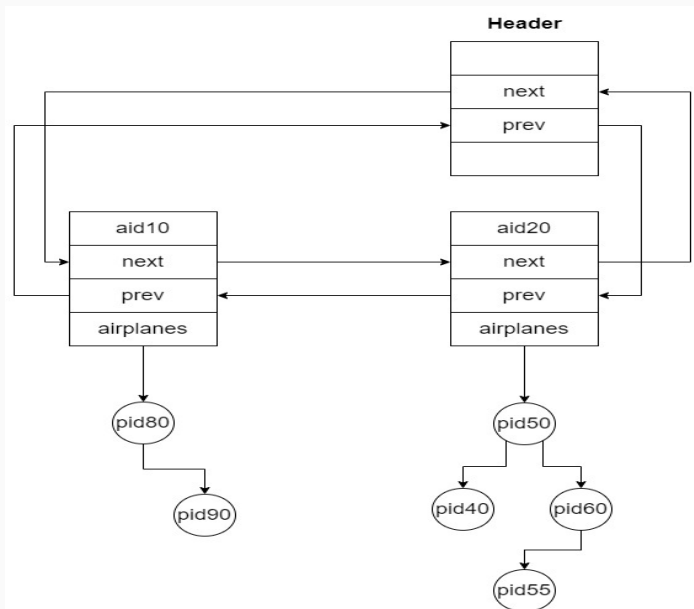
aid Αναγνωριστικό της αεροπορικής εταιρείας

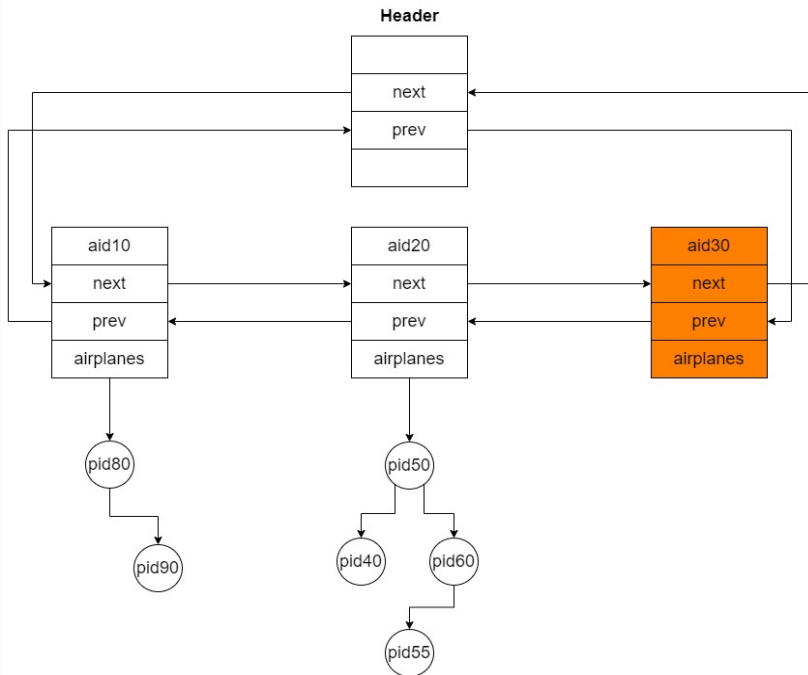
Κατά το γεγονός αυτό:

1. Θα γίνει εισαγωγή μίας νέας αεροπορικής εταιρείας με αναγνωριστικό [*aid*] στη λίστα αεροπορικών εταιρειών του ομίλου
2. **Προσοχή:** Μετά από κάθε εισαγωγή, η λίστα αεροπορικών εταιρειών θα πρέπει να παραμένει ταξινομημένη

Register Airline

R 30





Insert Airplane

Εισαγωγή ενός αεροπλάνου με αναγνωριστικό [*pid*] στο δέντρο αεροπλάνων της εταιρείας με αναγνωριστικό [*aid*] που πετάει προς τον προορισμό [*dest*] την ώρα [*depart_time*]

Insert Airplane

```
I pid aid dest depart_time
```

Ορίσματα :

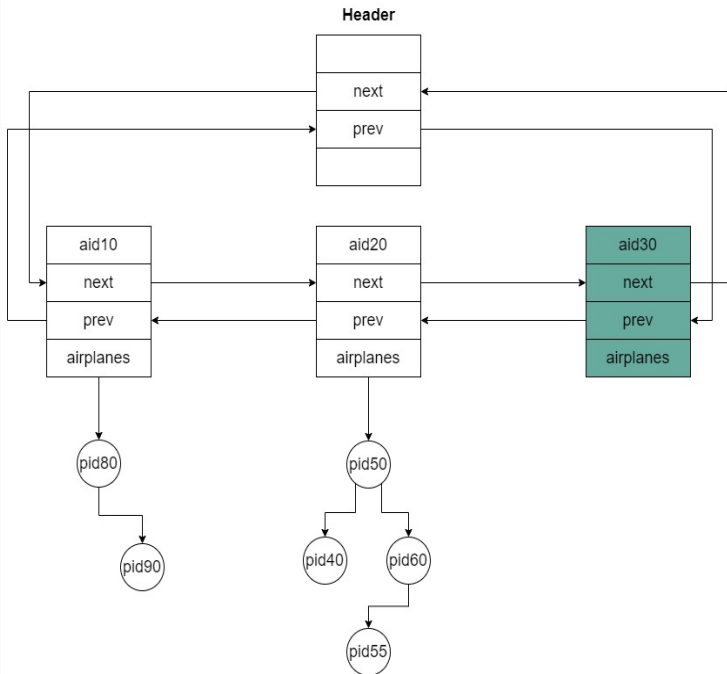
<i>pid</i>	Αναγνωριστικό αεροπλάνου
<i>aid</i>	Αναγνωριστικό αεροπορικής εταιρείας
<i>dest</i>	Αναγνωριστικό προορισμού
<i>depart_time</i>	Ώρα αναχώρησης του αεροπλάνου προς τον προορισμό

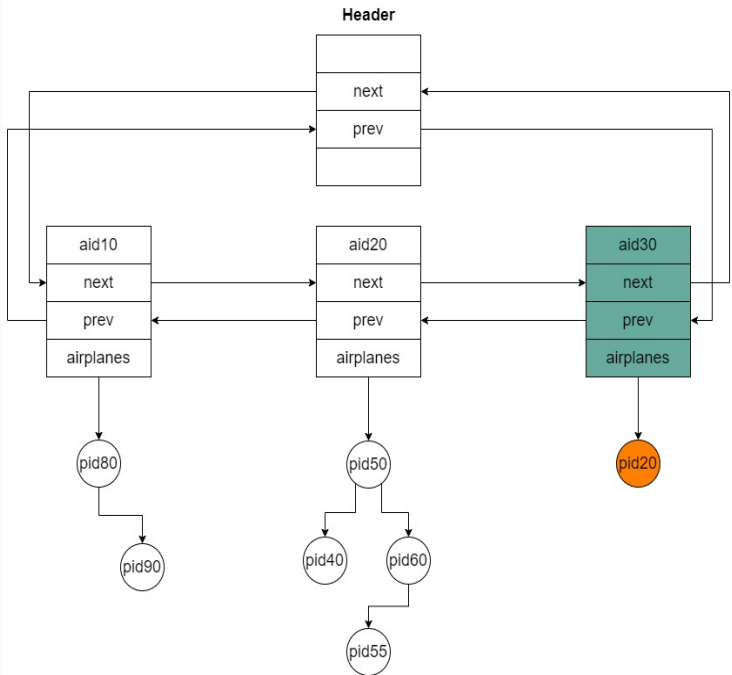
Κατά το γεγονός αυτό:

1. Αναζητούμε στη λίστα αεροπορικών εταιριών, την εταιρεία με αναγνωριστικό *aid*
2. Εισάγουμε στο δέντρο αεροπλάνων της εταιρείας ένα νέο κόμβο με αναγνωριστικό *pid*
3. Με την παράμετρο *dest* θα εντοπίσετε το κατάλληλο δέντρο πτήσεων στον πίνακα προορισμών και έπειτα θα εισάγετε σε αυτό ένα νέο κόμβο που να αντιπροσωπεύει τη νέα πτήση
4. **Προσοχή:** Μετά από κάθε εισαγωγή, το δέντρο αεροπλάνων και το δέντρο πτήσεων θα πρέπει να παραμένουν ταξινομημένα

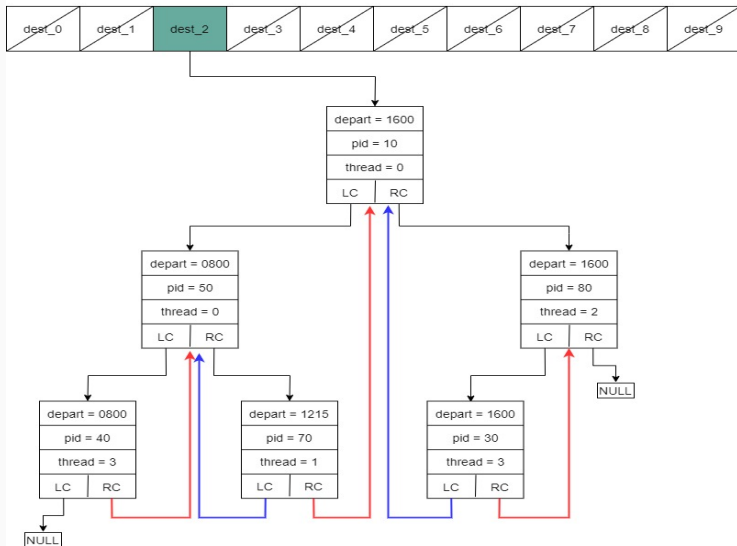
Insert Airplane

/ 20 30 2 1445

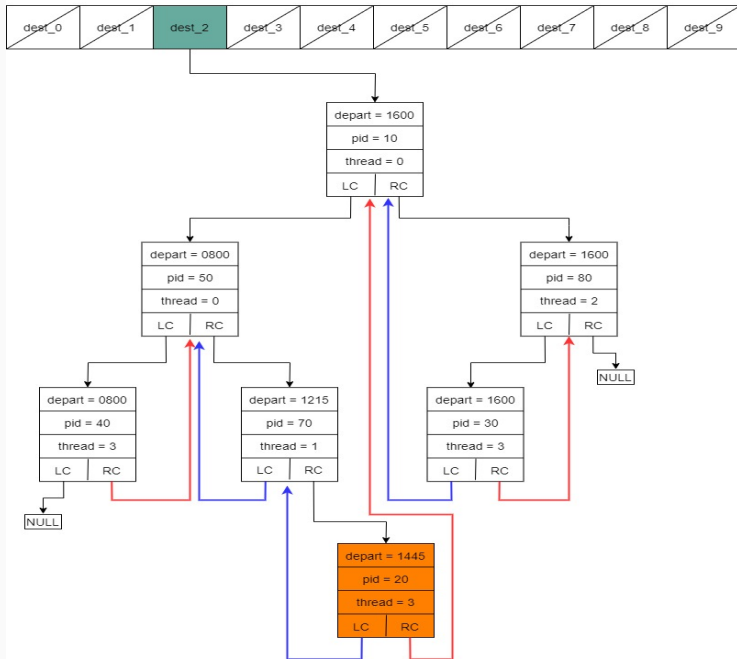




DESTINATIONS



DESTINATIONS



Cancel Flight

Απόσυρση ενός αεροπλάνου με αναγνωριστικό [*pid*] από το στόλο της αεροπορικής εταιρείας με αναγνωριστικό [*aid*] που ταξιδεύει στον προορισμό [*dest*]

Cancel Flight

C pid aid dest

Ορίσματα :

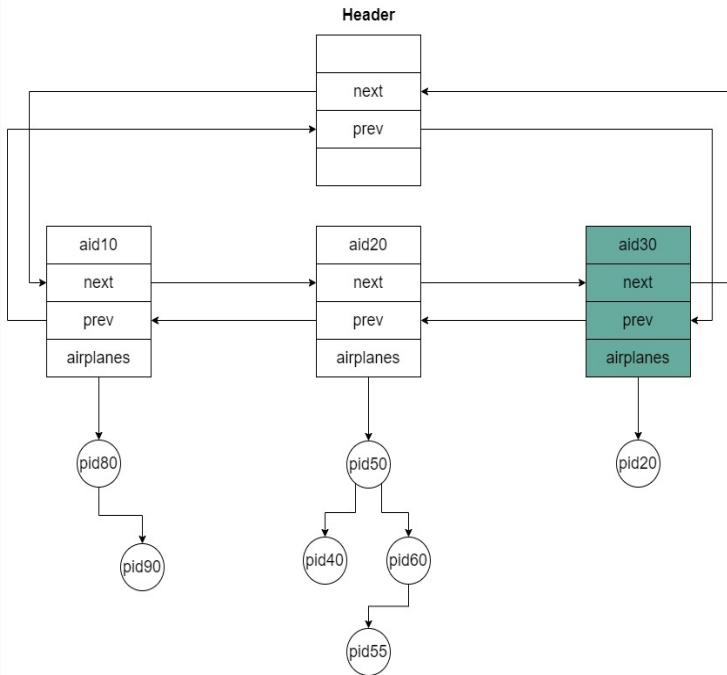
- pid* Αναγνωριστικό αεροπλάνου
- aid* Αναγνωριστικό αεροπορικής εταιρείας
- dest* Αναγνωριστικό προορισμού

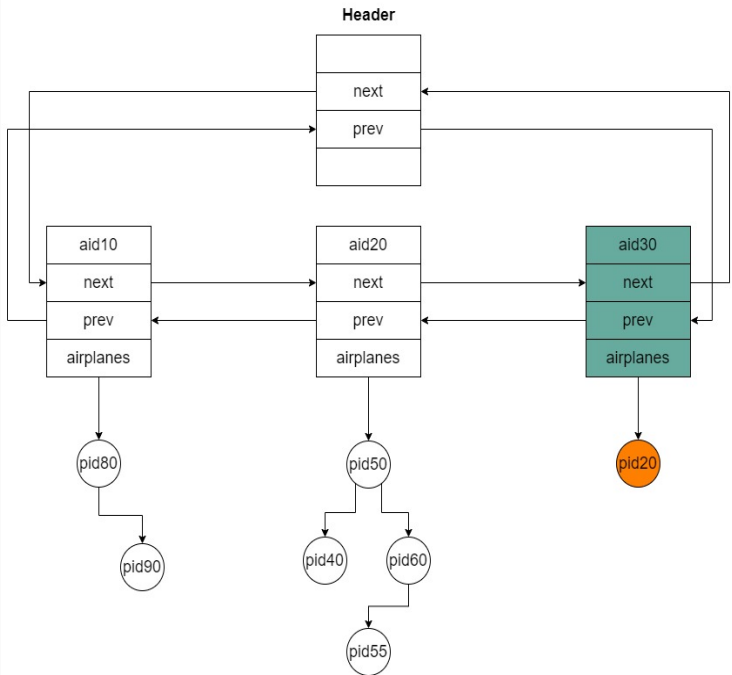
Κατά το γεγονός αυτό:

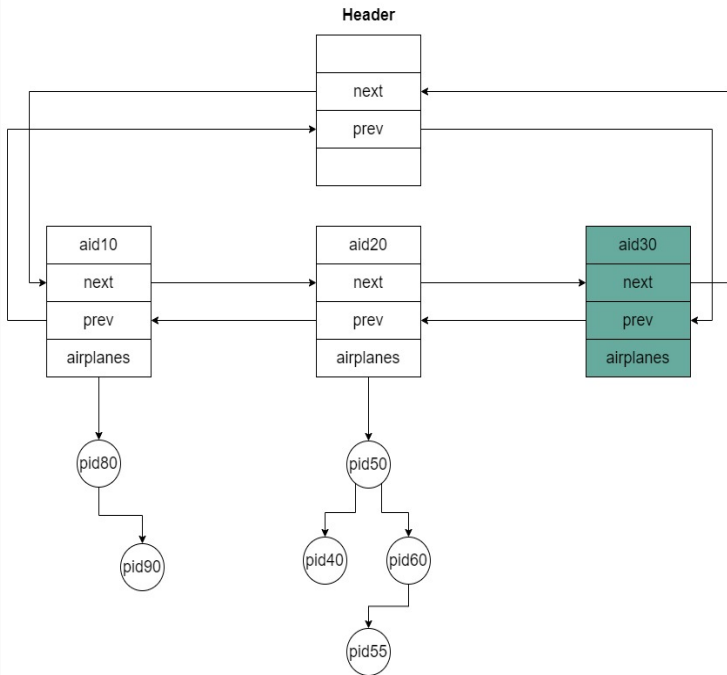
1. Αναζητάμε την εταιρεία με αναγνωριστικό *aid* στη λίστα αεροπορικών εταιρειών
2. Αφαιρούμε από τη λίστα αεροπλάνων της εταιρείας το αεροπλάνο με αναγνωριστικό *pid*
3. Εντοπίζουμε μέσω του *dest* το κατάλληλο δέντρο στον πίνακα προορισμών
4. Αφαιρούμε τον κόμβο με αναγνωριστικό *pid* από το δέντρο πτήσεων που αντιπροσωπεύει το αεροπλάνο

Cancel Flight

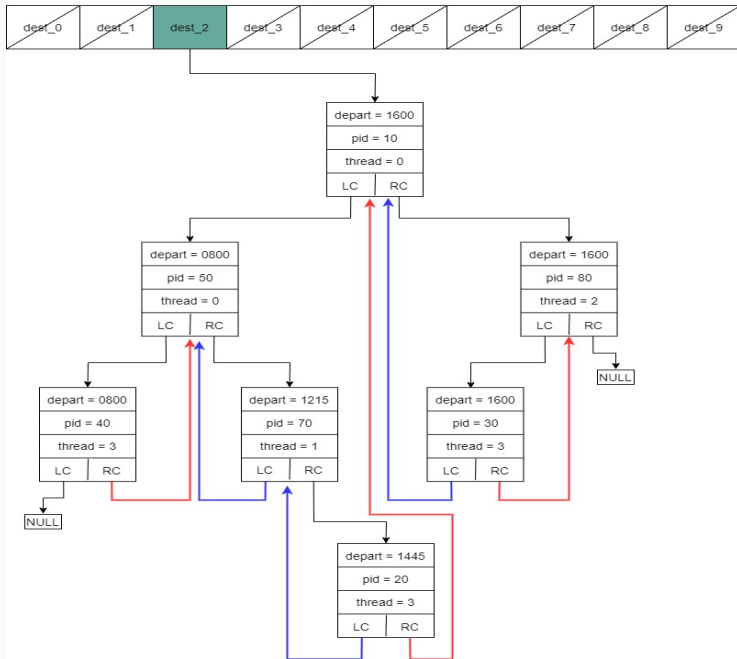
C 20 30 2



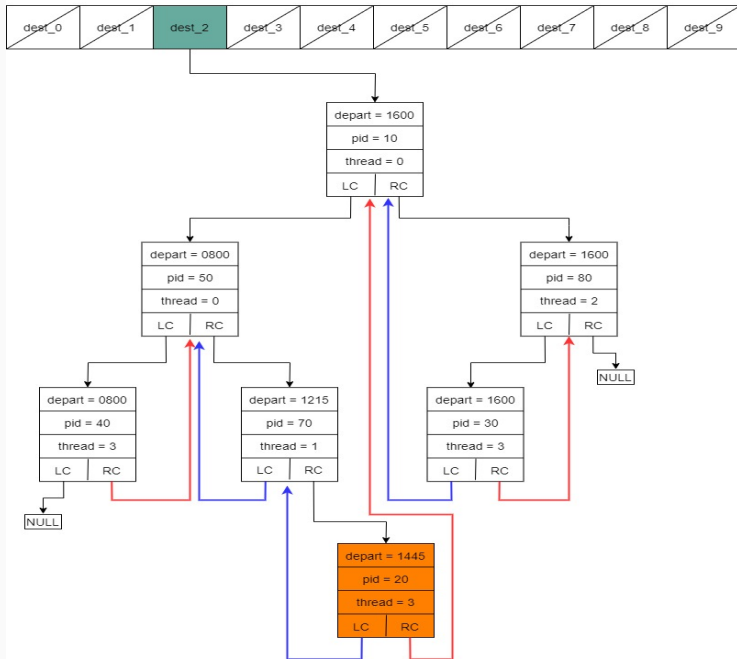




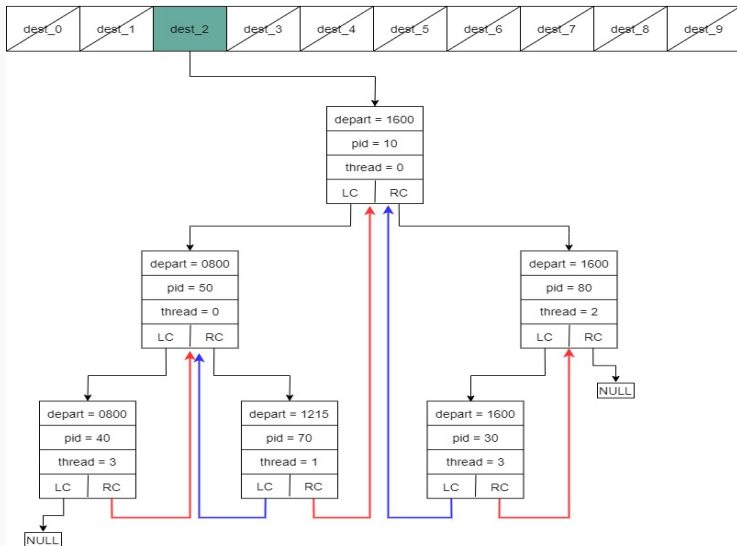
DESTINATIONS



DESTINATIONS



DESTINATIONS



Διαγραφή της εταιρείας με αναγνωριστικό [*aid*] από τον όμιλο

Delete Airline

D aid

Ορίσματα :

aid Αναγνωριστικό αεροπορικής εταιρείας

Κατά το γεγονός αυτό:

1. Αναζητούμε στη λίστα αεροπορικών εταιρειών την εταιρεία με αναγνωριστικό *aid*
2. Διατρέχουμε το δέντρο αεροπλάνων της εταιρείας και διαγράφουμε τον κάθε κόμβο
3. Για κάθε αεροπλάνο θα πρέπει:
 - 3.1 Χρησιμοποιώντας το *dest* του κάθε αεροπλάνου θα εντοπίζετε το δέντρο πτήσεων του προορισμού
 - 3.2 Χρησιμοποιώντας το *pid* θα εντοπίζετε και θα διαγράφετε το αεροπλάνο με αναγνωριστικό *pid* από το δέντρο πτήσεων του προορισμού
4. Θα πρέπει να διαγράφετε από τη λίστα αεροπορικών εταιρειών την εταιρεία με αναγνωριστικό *aid*

Εξαγορά της αεροπορικής εταιρείας $[aid_1]$ από την αεροπορική εταιρεία $[aid_2]$

Acquisition Airline

$A \text{ } aid_1 \text{ } aid_2$

Ορίσματα:

aid_1 Αναγνωριστικό αεροπορικής εταιρείας που εξαγοράζεται

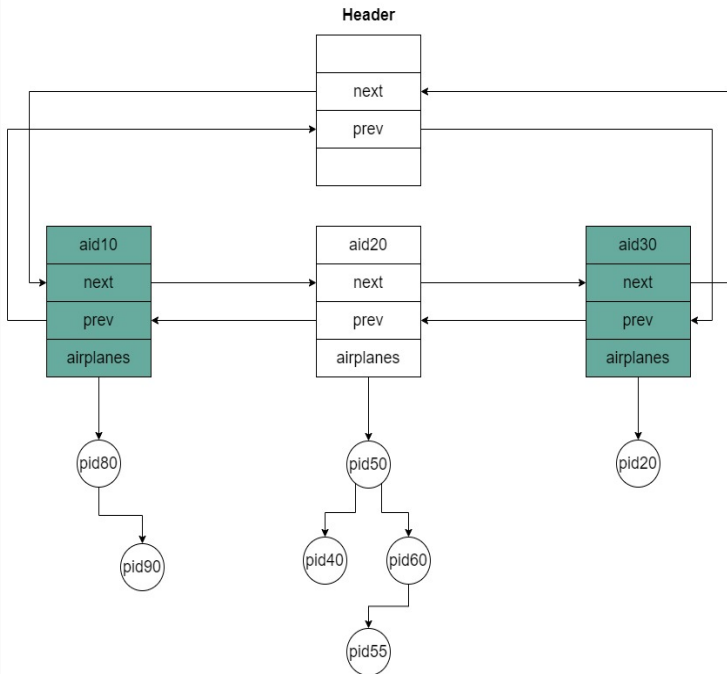
aid_2 Αναγνωριστικό αεροπορικής εταιρείας που εξαγοράζει

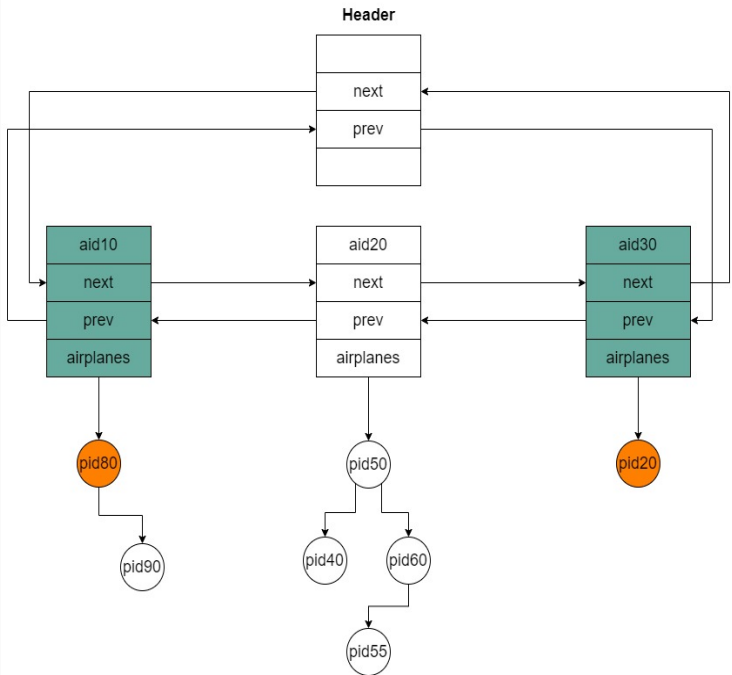
Κατά το γεγονός αυτό:

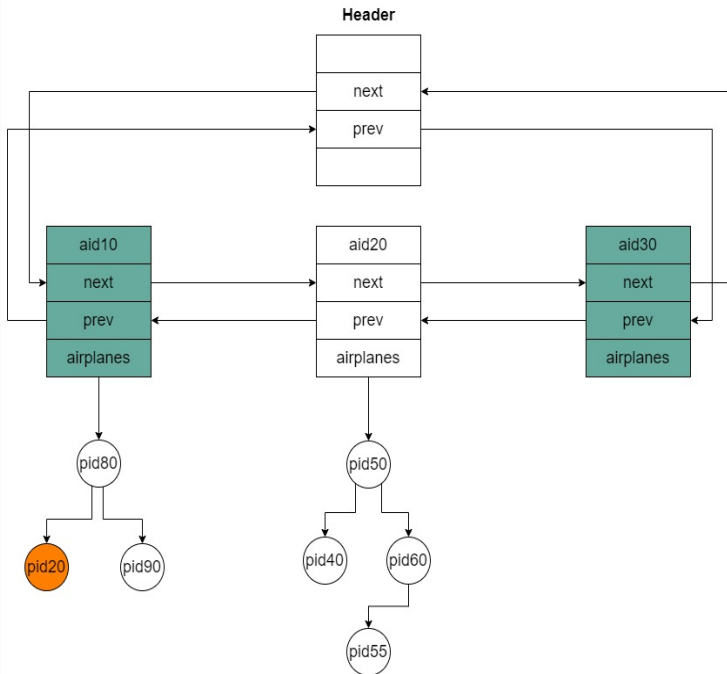
1. Αναζητάμε στη λίστα των αεροπορικών εταιρειών τις εταιρείες με αναγνωριστικά aid_1 και aid_2
2. Διατρέχουμε παράλληλα τα δέντρα αεροπλάνων των εταιρειών και τα συνενώνουμε
3. Διαγράφουμε την εταιρεία με αναγνωριστικό aid_1 από τη λίστα των αεροπορικών εταιρειών
4. **Προσοχή:** Το δέντρο των αεροπλάνων της αεροπορικής εταιρείας με αναγνωριστικό aid_2 που θα προκύψει μετά τη συνένωση θα πρέπει να είναι ταξινομημένο, ενώ το δέντρο αεροπλάνων της αεροπορικής εταιρείας aid_1 θα πρέπει να είναι κενό
5. **Προσοχή:** Η χρονική πολυπλοκότητα αυτού του γεγονότος θα πρέπει να είναι $O(n_1 + n_2)$, όπου n_1 και n_2 είναι το πλήθος στοιχείων των δέντρων αεροπλάνων των αεροπορικών εταιρειών

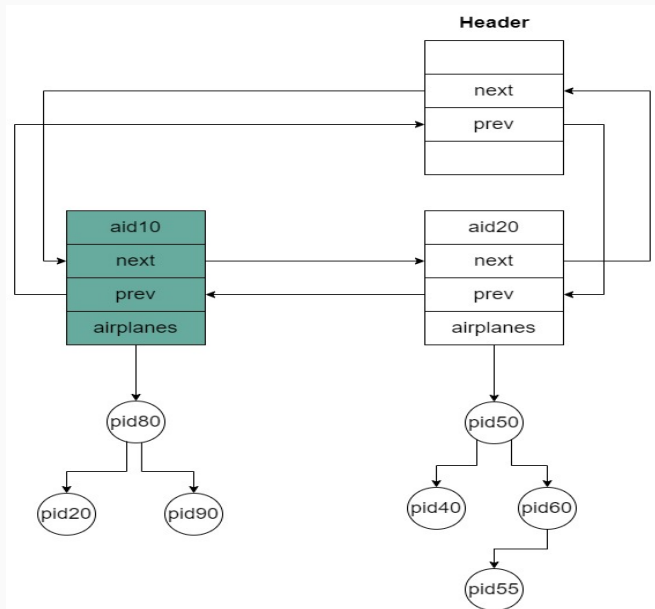
Acquisition Airline

A 30 10









Subsidiary Company

Η εταιρεία με αναγνωριστικό $[aid_2]$ εξαγοράζει τα αεροπλάνα της αεροπορικής εταιρείας με αναγνωριστικό $[aid_1]$, τα οποία εκτελούν πτήση προς ένα συγκεκριμένο προορισμό.

Subsidiary Company

$S \ aid_1 \ aid_2 \ dest$

Ορίσματα:

- aid_1 Αναγνωριστικό αεροπορικής εταιρείας που πουλά αεροπλάνα
- aid_2 Αναγνωριστικό αεροπορικής εταιρείας που εξαγοράζει αεροπλάνα
- $dest$ Αναγνωριστικό προορισμού

Κατά το γεγονός αυτό:

- Αναζητούμε στην λίστα των αεροπορικών εταιρειών τις εταιρείες με αναγνωριστικά aid_1 και aid_2 , αντίστοιχα
- Διατρέχουμε το δέντρο των αεροπλάνων της aid_1 , και βρίσκουμε κάθε αεροπλάνο που εκτελεί πτήση προς τον προορισμό $dest$
- Διαγράφουμε το αεροπλάνο από τη λίστα και το εισάγουμε στο δέντρο των αεροπλάνων της αεροπορικής εταιρείας με αναγνωριστικό aid_2
- **Προσοχή:** Το δέντρο των αεροπλάνων της εταιρείας με αναγνωριστικό aid_2 που θα προκύψει μετά την εκτέλεση του γεγονότος αυτού θα πρέπει να είναι ταξινομημένο
- **Προσοχή:** Η χρονική πολυπλοκότητα του γεγονότος θα πρέπει να είναι $O(m * h)$, όπου m είναι το πλήθος των αεροπλάνων που εξαγοράζονται και h το ύψος του δέντρου αεροπλάνων της αεροπορικής εταιρείας που εξαγοράζει τα αεροπλάνα

Ένας πελάτης ψάχνει όλες τις πτήσεις που αναχωρούν ακριβώς μετά την ώρα $[ts]$ με προορισμό $[dest]$

Find Flight

$F \text{ } dest \text{ } ts$

Ορίσματα:

- $dest$ Αναγνωριστικό προορισμού
- ts Αναγνωριστικό ώρας αναχώρησης

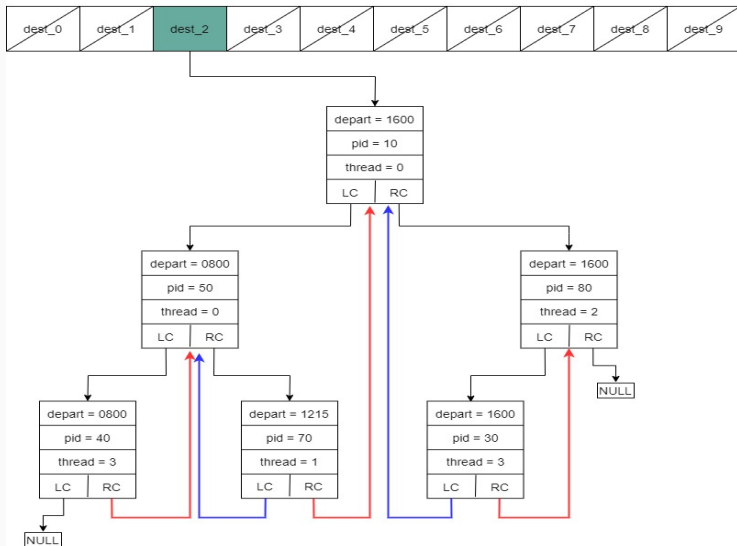
Κατά το γεγονός αυτό:

- Διατρέχουμε μέρος του δένδρου πτήσεων του προορισμού *dest*, αναζητώντας αρχικά τον κόμβο με χρονοσφραγίδα *ts*
- Στη συνέχεια με τη χρήση των νημάτων διατρέχουμε το δέντρο πτήσεων και εκτυπώνουμε τις πτήσεις που αναχωρούν μετά την ώρα *ts*

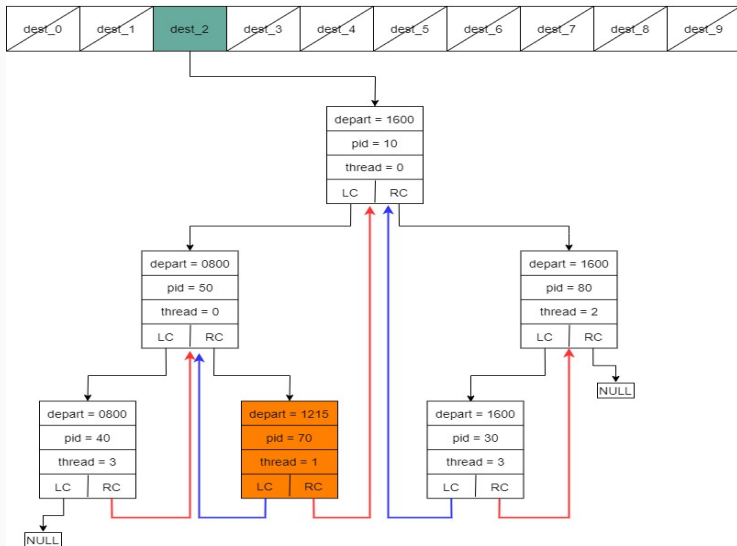
Find Flight

F 2 1215

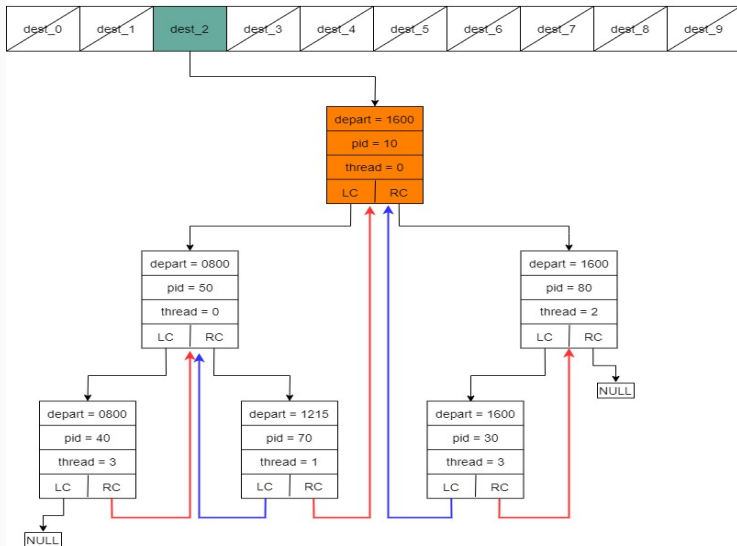
DESTINATIONS



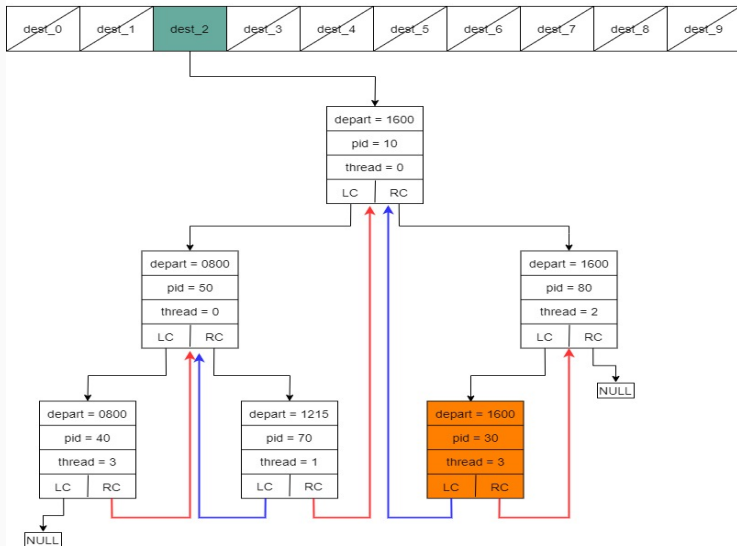
DESTINATIONS



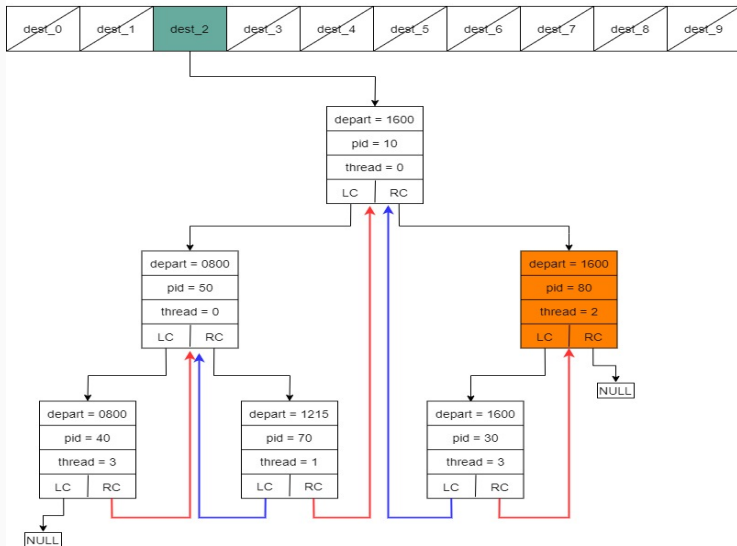
DESTINATIONS



DESTINATIONS



DESTINATIONS



Ένας νέος χρήστης με αναγνωριστικό [*cid*] εγγράφεται στο σύστημα ως πελάτης του ομίλου με μηδενικά μίλια.

New Client

N cid

Ορίσματα:

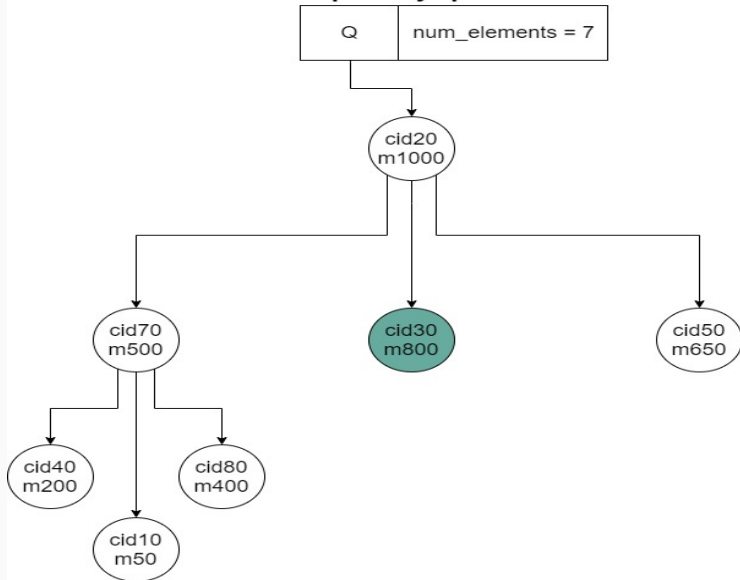
cid Αναγνωριστικό πελάτη

- Χρησιμοποιούμε το πεδίο *num_elements* και το ότι το δέντρο είναι πλήρες για να βρούμε τον κόμβο στον οποίο πρέπει να τοποθετηθεί ο νέος κόμβος ως παιδί.
- **Προσοχή:** Μετά την εισαγωγή, ίσως να χρειάζεται να επανορθώσουμε την ιδιότητα της μερικής διάταξης στο δέντρο, ανταλλάσσοντας τον κόμβο που μόλις εισάγαμε με τον γονικό του κόμβο. Η διαδικασία αυτή μπορεί να πρέπει να επαναληφθεί μέχρι ο νέος κόμβος να βρεθεί στη θέση που πρέπει, σύμφωνα με τα μίλια του.
- **Προσοχή:** Η χρονική πολυπλοκότητα του γεγονότος θα πρέπει να είναι $O(h)$, όπου h είναι το ύψος του δένδρου που αναπαριστά την ουρά προτεραιότητας.

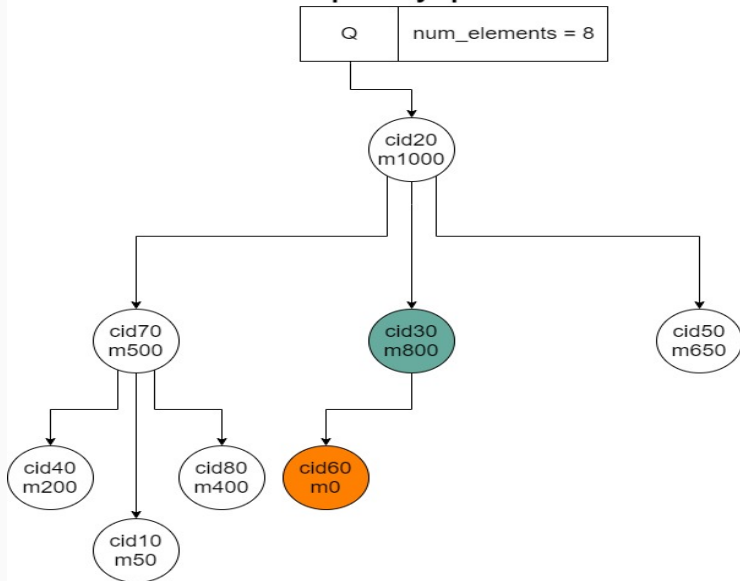
New Client

$N = 60$

priority queue



priority queue



Ο πελάτης με τη μεγαλύτερη προτεραιότητα διαγράφεται από το σύστημα.

Erase Client

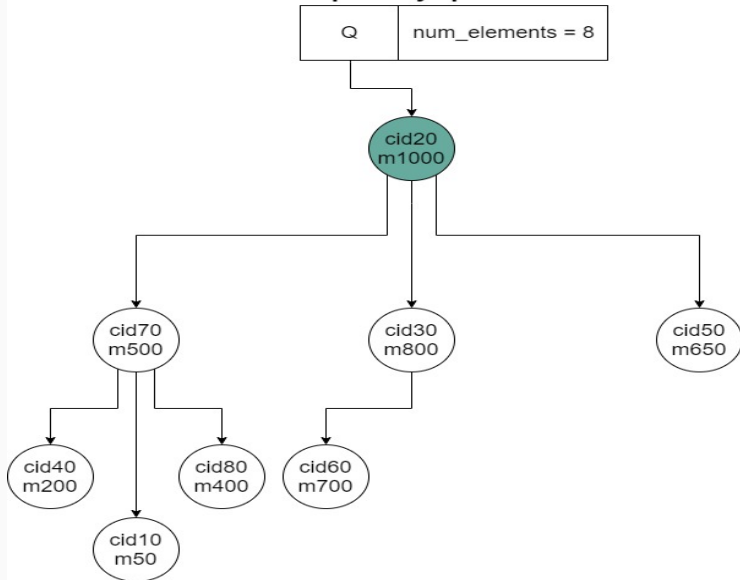
E

- Ο πελάτης με τη μέγστη προτεραιότητα βρίσκεται στη ρίζα. Για να τον διαγράψουμε θα πρέπει να πάρει τη θέση του ένα από τα παιδιά του.
- **Προσοχή:** Μετά τη διαγραφή θα πρέπει να επανορθώσουμε την ιδιότητα της μερικής διάταξης στο δέντρο, εκτελώντας ό,τι αλλαγές χρειάζονται ώστε το δένδρο να συνεχίσει να αναπαριστά ουρά προτεραιότητας μεγίστου.

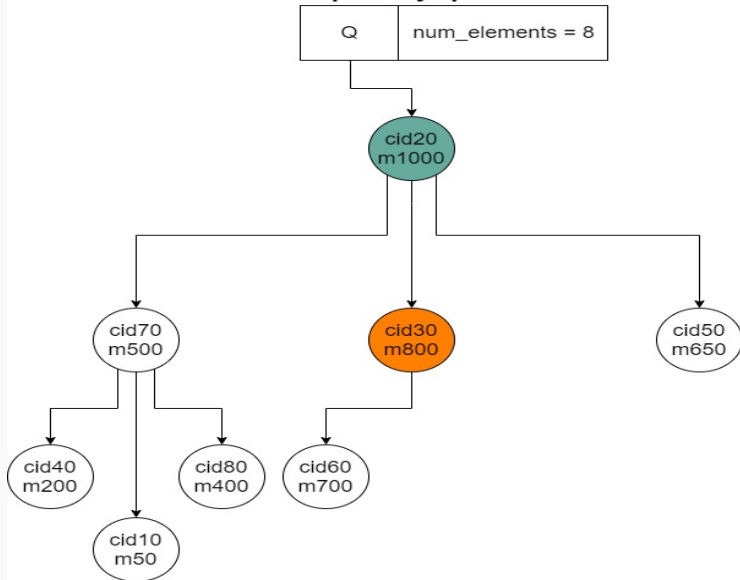
Erase Client

E

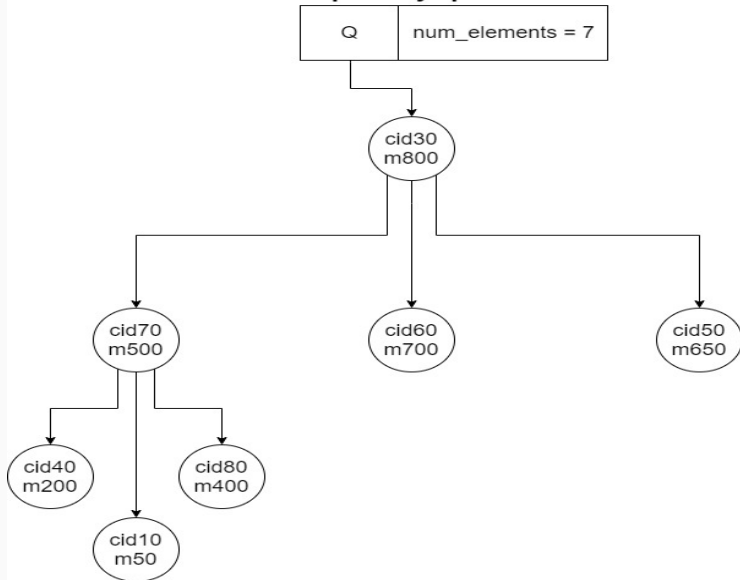
priority queue



priority queue



priority queue



Ένας πελάτης με αναγνωριστικό [*cid*] ταξιδεύει στον προορισμό με αναγνωριστικό [*dest*].

Travel Client

T cid dest

Ορίσματα:

cid Αναγνωριστικό πελάτη
dest Αναγνωριστικό προορισμού

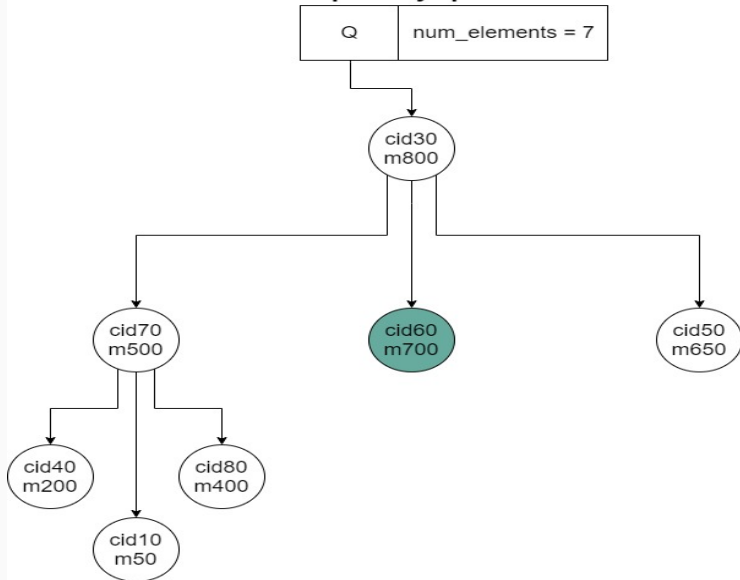
Κατά το γεγονός αυτό:

- Αναζητούμε τον πελάτη *cid* στο δέντρο πελατών
- Αφού τον βρούμε, αυξάνουμε τα μίλια του ανάλογα
- Πραγματοποιούμε ό,τι αλλαγές χρειάζονται ώστε το δέντρο να συνεχίσει να αναπαριστά ουρά προτεραιότητας μεγίστου.

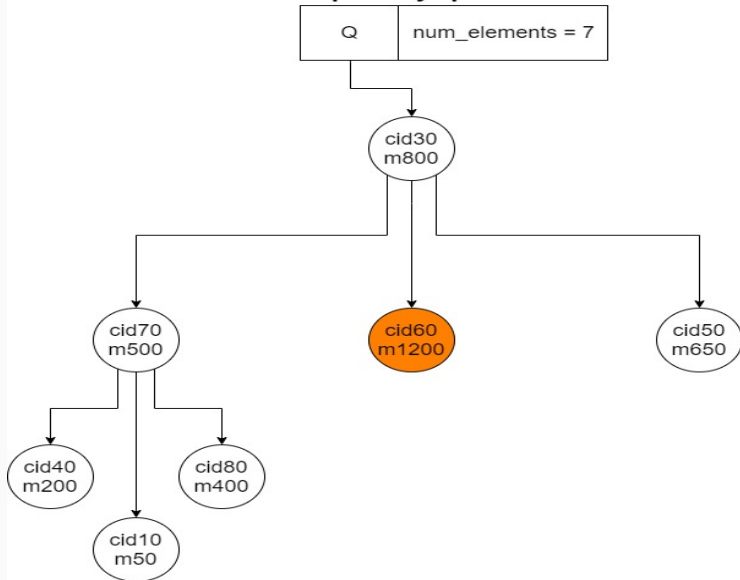
Travel Client

T 60 4

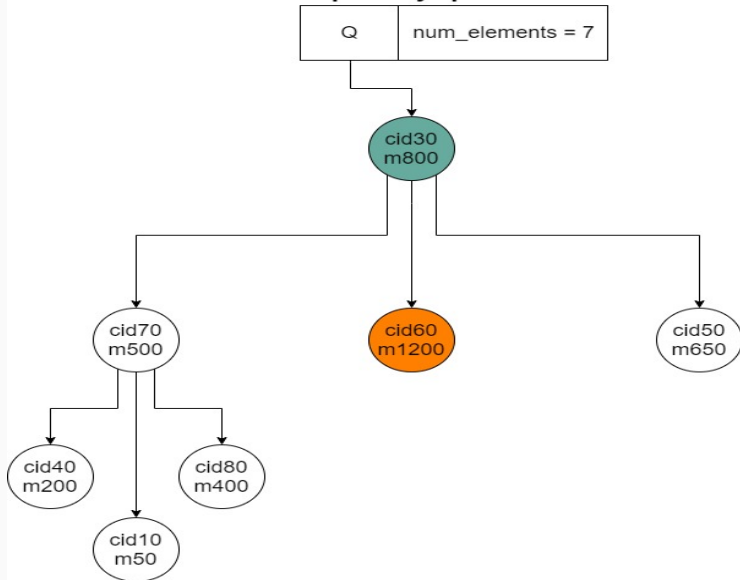
priority queue



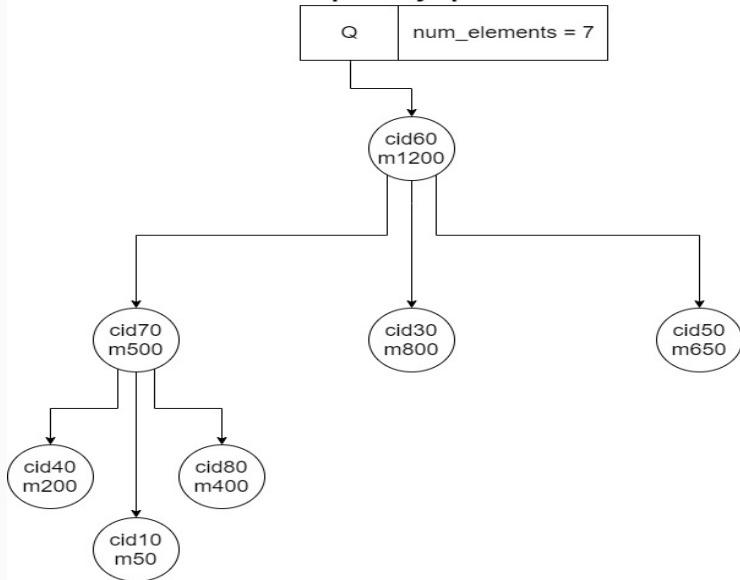
priority queue



priority queue



priority queue



Εκτύπωση όλων των αεροπορικών εταιρειών

Print Airlines

X

Εκτύπωση όλων των προορισμών από τον πίνακα προορισμών

Print Destinations

Υ

Εκτύπωση κατά επίπεδα όλων των πελατών από το δέντρο πελατών.

Print Clients

Z

- Πρέπει να υλοποιήσουμε διάσχιση κατά επίπεδα χρησιμοποιώντας μία ουρά FIFO.
- Κατά την εκτέλεση του γεγονότος γνωρίζουμε τα πλήθος των στοιχείων στην ουρά προτεραιότητας, άρα η ουρά FIFO μπορεί να υλοποιηθεί με στατικό τρόπο (με πίνακα).

Thanks!