

## **ΗΥ240: Δομές Δεδομένων**

### **Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2015-16**

#### **Διδάσκουσα: Παναγιώτα Φατούρου**

#### **Προγραμματιστική Εργασία - 2<sup>ο</sup> Μέρος**

**Ημερομηνία Παράδοσης:** Παρασκευή, 18 Δεκεμβρίου 2015, ώρα 23:59.

**Τρόπος Παράδοσης:** Χρησιμοποιώντας το πρόγραμμα turnin. Πληροφορίες για το πώς λειτουργεί το turnin παρέχονται στην ιστοσελίδα του μαθήματος.



#### **Γενική Περιγραφή**

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που προσομοιώνει τη **διαδικασία επιλογής και αποστολής δώρων του Αγίου Βασιλείου** σε γεωγραφικά διαμερίσματα διάφορων χωρών του κόσμου κατά την περίοδο των Χριστουγέννων.

### Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Για την υλοποίηση της προσομοίωσης της διαδικασίας επιλογής και αποστολής δώρων του Αγίου Βασιλείου, παιδιά από γεωγραφικά διαμερίσματα διάφορων χωρών του κόσμου στέλνουν γράμματα με τα στοιχεία και τις προτιμήσεις τους.

Για την καλύτερη διαχείριση των γραμμάτων, τα παιδιά πρέπει να διαχωριστούν σε κατηγορίες με βάση την ηλικία τους. Πιο συγκεκριμένα, υπάρχουν τέσσερις ηλικιακές ομάδες: η πρώτη ομάδα περιέχει παιδιά ηλικίας 0-3 ετών, η δεύτερη 4-7 ετών, η τρίτη 8-11 ετών και η τέταρτη 12-15 ετών. Για τη διαχείριση των ηλικιακών ομάδων, θα δημιουργήσετε έναν πίνακα σταθερού μεγέθους **4 θέσεων**, ο οποίος θα ονομάζεται **πίνακας ηλικιακών κατηγοριών**. Η κάθε θέση του πίνακα θα περιέχει ένα δείκτη (τύπου child) στη ρίζα ενός απλά συνδεδεμένου, μη-ταξινομημένου δυαδικού δέντρου, το οποίο θα ονομάζεται **δέντρο παιδιών της ηλικιακής κατηγορίας** αυτής. Ο κάθε κόμβος του δέντρου περιέχει πληροφορίες για κάποιο παιδί που ανήκει σε αυτήν την ηλικιακή κατηγορία. Ένας τέτοιος κόμβος θα αποτελεί μια εγγραφή τύπου child με τα ακόλουθα πεδία:

- **cid**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το παιδί.
- **age**: Αριθμός (τύπου int) που αντιστοιχεί στην ηλικία του παιδιού.
- **did**: Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το γεωγραφικό διαμέρισμα στο οποίο κατοικεί το παιδί (περισσότερες λεπτομέρειες για τα γεωγραφικά διαμερίσματα παρακάτω).
- **present\_choices**: Ένας πίνακας σταθερού μεγέθους τριών (3) θέσεων με τις προτιμήσεις δώρων του παιδιού. Το κάθε κελί αυτού του πίνακα περιέχει μοναδικά αναγνωριστικά δώρων (τύπου int), σύμφωνα με τις προτιμήσεις του παιδιού.
- **rc**: Δείκτης (τύπου child) στο δεξί παιδί του κόμβου
- **lc**: Δείκτης (τύπου child) στο αριστερό παιδί του κόμβου

Το δέντρο παιδιών κάθε ηλικιακής κατηγορίας είναι **μη-ταξινομημένο δυαδικό δέντρο**. Η εισαγωγή ενός κόμβου με αναγνωριστικό cid σε ένα τέτοιο δένδρο θα γίνεται με τον παρακάτω τρόπο:

- 1.1. Εφαρμόζοντας κάποιου είδους διάσχιση στο δέντρο, θα εντοπίζετε τον κόμβο με το αμέσως μικρότερο αναγνωριστικό από το cid. Για παράδειγμα, για να εισάγουμε ένα νέο παιδί με αναγνωριστικό 22 σε ένα δέντρο που περιέχει ήδη παιδιά με αναγνωριστικά: 10, 15, 20, 18, 30, 25, ο ζητούμενος κόμβος είναι αυτός με αναγνωριστικό 20.
- 1.2. Αν δεν υπάρχει κόμβος με μικρότερο αναγνωριστικό από τον κόμβο προς εισαγωγή, τότε θα εντοπίσετε τον κόμβο του δέντρου με το μικρότερο αναγνωριστικό παιδιού.
2. Αν ο κόμβος, nd, που εντοπίσατε έχει το πολύ ένα παιδί, θα εισάγεται τον νέο κόμβο ως παιδί του κόμβου αυτού (αριστερό αν ο κόμβος nd δεν έχει αριστερό παιδί και δεξιό διαφορετικά).
3. Αν ο nd έχει δύο παιδιά, τότε θα βρίσκετε τον επόμενο ή τον προηγούμενο του nd στην ενδοδιατεταγμένη διάσχιση και θα εισάγετε το νέο κόμβο ως παιδί του κόμβου αυτού (αριστερό αν ο κόμβος δεν έχει αριστερό παιδί και δεξιό διαφορετικά).

Στο Σχήμα 1 παρουσιάζεται ένα παράδειγμα της μεθόδου που περιγράφεται παραπάνω. Στα παραδείγματα των Σχημάτων 1(b) και 1(c), ο εντοπισμός των κόμβων με το αμέσως μικρότερο αναγνωριστικό από το προς εισαγωγή αναγνωριστικό, γίνεται σύμφωνα με το βήμα 1.2. Στο παράδειγμα του Σχήματος 1(d), ο

κόμβος με το αμέσως μικρότερο αναγνωριστικό από το 23 είναι ο κόμβος με αναγνωριστικό 20. Ο κόμβος αυτός έχει δύο παιδιά. Γι' αυτό επιλέγουμε, ως γονικό κόμβο του κόμβου με κλειδί 23, τον προηγούμενο του κόμβου με κλειδί 20 στην ενδοδιατεταγμένη διάσχιση, όπως περιγράφεται στο βήμα 3. Στη συνέχεια, για την εισαγωγή του κόμβου με αναγνωριστικό 22, ο κόμβος με το αμέσως μικρότερο κλειδί είναι πάλι αυτός με αναγνωριστικό 20, ο οποίος έχει δύο παιδιά. Αυτή τη φορά, επιλέγεται ο επόμενος κόμβος του κόμβου με αναγνωριστικό 20 στην ενδοδιατεταγμένη διάσχιση ως γονικός κόμβος του κόμβου με αναγνωριστικό 22, όπως φαίνεται στο Σχήμα 1(e). Τέλος στο παράδειγμα του Σχήματος 1(f), η εισαγωγή του κόμβου με κλειδί 12 γίνεται με τον τρόπο που περιγράφεται στο βήμα 1.1

Για τη διαγραφή κάποιου κόμβου θα εφαρμοστεί η ίδια μέθοδος που ακολουθείται στα δυαδικά δέντρα αναζήτησης. Πιο συγκεκριμένα θα πρέπει να αναγνωρίζετε τις παρακάτω περιπτώσεις:

- Ο προς διαγραφή κόμβος είναι φύλλο. Διαγράφουμε τον κόμβο χωρίς να εκτελέσουμε κάποια επιπρόσθετη ενέργεια.
- Ο προς διαγραφή κόμβος είναι εσωτερικός αλλά έχει μόνο ένα παιδί. Αντικαθιστούμε τον κόμβο με το μοναδικό παιδί του.
- Ο προς διαγραφή κόμβος είναι εσωτερικός με δύο παιδιά. Αντικαθιστούμε τον κόμβο με τον επόμενο ή τον προηγούμενό του στην ενδο-διατεταγμένη διάσχιση.

Στο Σχήμα 2 παρουσιάζεται ο πίνακας ηλικιακών κατηγοριών, σταθερού μεγέθους 4 θέσεων, και το δέντρο των παιδιών που δεικτοδοτείται από το εκάστοτε στοιχείο του. Στο σχήμα 3 παρουσιάζεται αναλυτικότερα η δομή του δέντρου παιδιών μιας ηλικιακής κατηγορίας.

Ο Άγιος Βασίλειος διατηρεί μια αποθήκη με αποθέματα δώρων. Για το σκοπό αυτό θα δημιουργήσετε ένα ταξινομημένο, διπλά-συνδεδεμένο, δυαδικό δέντρο με κόμβο φρουρό το οποίο θα ονομάζεται **δέντρο αποθεμάτων**. Κάθε κόμβος της λίστας αποθεμάτων θα περιέχει μια εγγραφή τύπου `present` με τα ακόλουθα πεδία:

- **pid**: Αναγνωριστικό (τύπου `int`) που χαρακτηρίζει μοναδικά το δώρο.
- **stock\_cnt**: Αριθμός (τύπου `int`) που αντιστοιχεί στο απόθεμα του δώρου που είναι διαθέσιμο στην αποθήκη.
- **request\_cnt**: Αριθμός (τύπου `int`) που αντιστοιχεί στο πλήθος των παιδιών που έχουν εκφράσει επιθυμία για το δώρο. Αρχικά, ο μετρητής αυτός θα έχει τιμή 0 και θα αυξάνεται καθώς ο Άγιος Βασίλειος δέχεται τα γράμματα των παιδιών.
- **parent**: Δείκτης (τύπου `present`) στον πατρικό κόμβο.
- **lc**: Δείκτης (τύπου `present`) στο αριστερό παιδί του κόμβου.
- **rc**: Δείκτης (τύπου `present`) στο δεξί παιδί του κόμβου.

Το δέντρο αυτό θα είναι ταξινομημένο ως προς το αναγνωριστικό του κάθε δώρου. Ο κόμβος φρουρός είναι και αυτός τύπου `present` με την ιδιαιτερότητα ότι το αναγνωριστικό του δώρου (`pid`) και οι μετρητές του (`stock_cnt`, `request_cnt`) έχουν τιμή -1 ενώ οι δείκτες (`parent`, `lc`, `rc`) έχουν τιμή `NULL`.

Το Σχήμα 4 απεικονίζει το δέντρο αποθεμάτων των δώρων (`presents`). Στο δέντρο αυτό εφαρμόζεται το ακόλουθο ευριστικό αναδιάρθρωσης του δέντρου κάθε φορά που **αναζητούμε** έναν κόμβο: ο κόμβος που αναζητήθηκε θα μεταφέρεται στη ρίζα του δέντρου. Αυτό θα πρέπει να γίνεται κάνοντας κατάλληλες

περιστροφές ώστε να μην καταστρέφεται η ιδιότητα της ταξινόμησης του δένδρου. Το ευριστικό αυτό αποσκοπεί στη διατήρηση των κόμβων που αναζητούνται πιο συχνά, κοντά στη ρίζα του δέντρου και άρα εφαρμόζεται για λόγους απόδοσης. Το ευριστικό εφαρμόζεται μόνο κάθε φορά που γίνεται αναζήτηση για κάποιο αναγνωριστικό δώρου στο δένδρο.

Για να ικανοποιήσει τις επιθυμίες των παιδιών, ο Άγιος Βασίλειος διαχωρίζει τις παραδόσεις των δώρων με βάση το γεωγραφικό διαμέρισμα (και άρα και τη χώρα) που διαμένει το κάθε παιδί. Για αυτό το σκοπό θα χρησιμοποιήσετε μια **ταξινομημένη, απλά-συνδεδεμένη** λίστα, η οποία θα ονομάζεται **λίστα γεωγραφικών διαμερισμάτων** (DistrictsL). Η λίστα αυτή είναι ταξινομημένη βάσει του αναγνωριστικού του γεωγραφικού διαμερίσματος. Κάθε στοιχείο της **λίστας γεωγραφικών διαμερισμάτων** είναι μια εγγραφή τύπου district με τα ακόλουθα πεδία:

- **did:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά ένα γεωγραφικό διαμέρισμα και τη χώρα στην οποία ανήκει.
- **Children\_cnt:** Αριθμός (τύπου int) που αντιστοιχεί στον αριθμό των παιδιών που διαμένουν στο γεωγραφικό διαμέρισμα με αναγνωριστικό did.
- **assignHT:** Δείκτης προς έναν πίνακα κατακερματισμού (hash table), ο οποίος ονομάζεται **πίνακας κατακερματισμού ανάθεσης δώρων** του γεωγραφικού διαμερίσματος. Στον πίνακα αυτό θα αποθηκεύονται τα δώρα που ανατίθενται στα παιδιά που διαμένουν σ' αυτό το γεωγραφικό διαμέρισμα. Ο πίνακας κατακερματισμού θα είναι σταθερού μεγέθους και θα καθορίζεται με βάση τον αριθμό των παιδιών που ανήκουν στο κάθε γεωγραφικό διαμέρισμα. Στη δομή αυτή (πίνακας κατακερματισμού) θα αποθηκεύετε στοιχεία τύπου present\_assignment με τα ακόλουθα πεδία:
  - **cid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το παιδί.
  - **pid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το δώρο.
  - **s\_degree:** Αριθμός (τύπου int) που αντιστοιχεί στο βαθμό ικανοποίησης του παιδιού. Η μεταβλητή αυτή λαμβάνει τιμές από 1 μέχρι 5, με 1 να αντιστοιχεί στον ελάχιστο βαθμό ικανοποίησης του παιδιού και 5 στο μέγιστο βαθμό ικανοποίησης.
- **next:** Δείκτης (τύπου district) στον επόμενο κόμβο της **λίστας γεωγραφικών διαμερισμάτων**.

Για τη διαχείριση των συγκρούσεων στους πίνακες κατακερματισμού των διαφορετικών γεωγραφικών διαμερισμάτων θα υλοποιήσετε τρεις διαφορετικές μεθόδους τις οποίες έχετε διδαχθεί στα πλαίσια του μαθήματος:

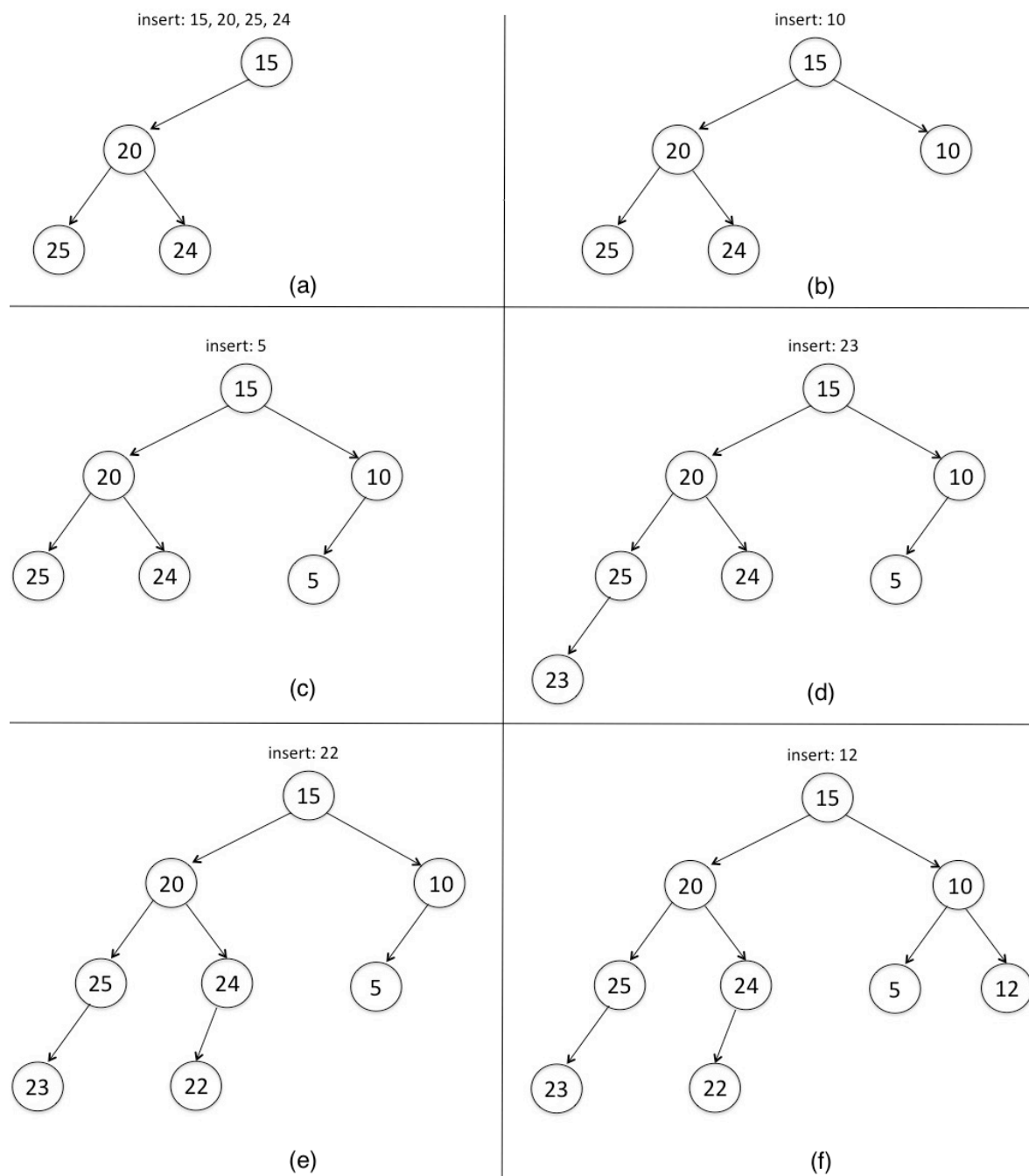
- Μεικτές αλυσίδες με κελάρι
- Διπλός κατακερματισμός
- Ταξινομημένος διπλός κατακερματισμός

Συγκεκριμένα, ο πίνακας κατακερματισμού κάθε γεωγραφικού διαμερίσματος θα πρέπει να υλοποιηθεί βάσει **μιας εκ των παραπάνω μεθόδων**. Για να καθορίσετε τη μέθοδο που θα εφαρμόσετε στον πίνακα κατακερματισμού που αντιστοιχεί στο γεωγραφικό διαμέρισμα με αναγνωριστικό did, θα εξετάσετε το υπόλοιπο της διαίρεσης του did με το 3 (δηλαδή το αποτέλεσμα του  $\text{did} \bmod 3$ ). Αν η τιμή του υπολοίπου αυτού είναι:

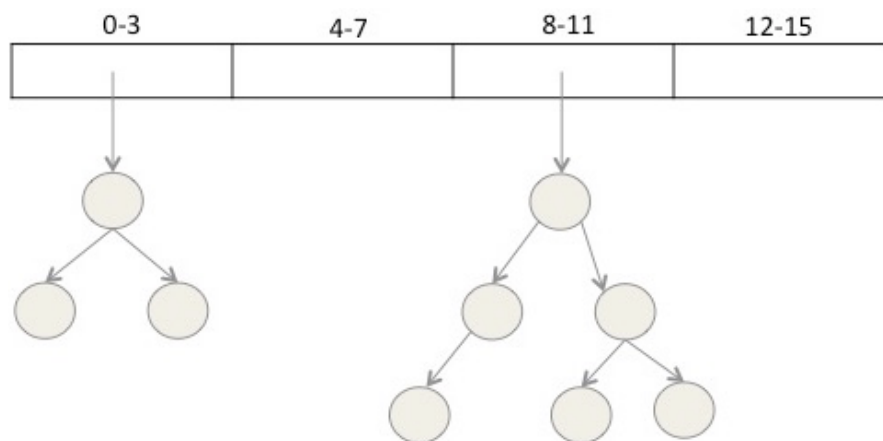
- 0, θα εφαρμοστεί η πρώτη μέθοδος (μεικτές αλυσίδες με κελάρι)
- 1, θα εφαρμοστεί η δεύτερη μέθοδος (διπλός κατακερματισμός)
- 2, θα εφαρμοστεί η τρίτη μέθοδος (ταξινομημένος διπλός κατακερματισμός)

Για την υλοποίηση της συνάρτησης κατακερματισμού θα πρέπει να χρησιμοποιήσετε καθολικό κατακερματισμό. Για την υλοποίηση του καθολικού κατακερματισμού παρέχεται, ένας πίνακας `primes_g[]` με 160 πρώτους αριθμούς σε αύξουσα σειρά. Επίσης απαιτείται το μέγιστο κλειδί (αναγνωριστικό) που έχει οποιοδήποτε παιδί, το οποίο μπορείτε να εντοπίσετε καθώς ο άγιος Βασίλειος λαμβάνει γράμματα από τα παιδιά όπως περιγράφεται παρακάτω.

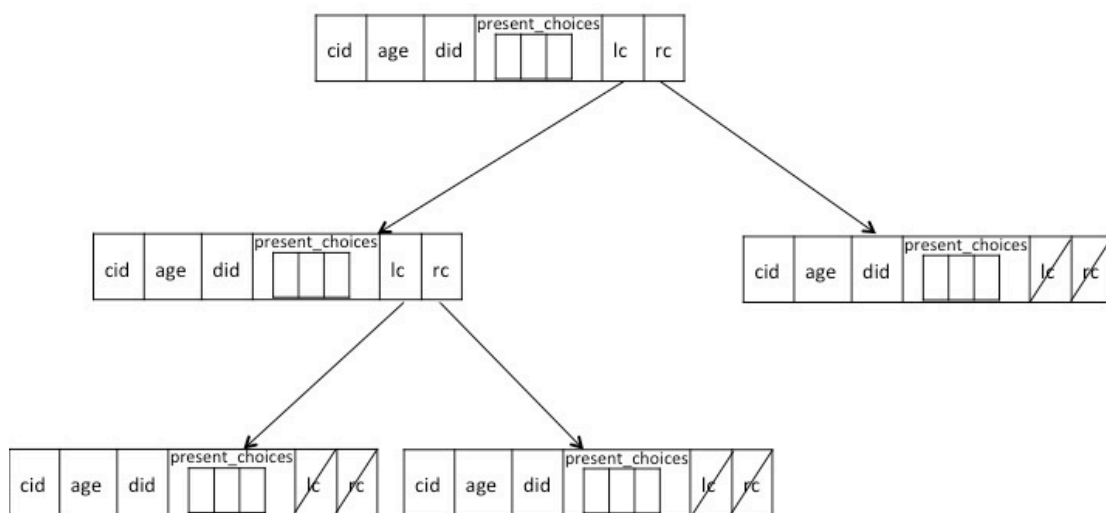
Στο Σχήμα 5 παρουσιάζεται σχηματικά η λίστα γεωγραφικών διαμερισμάτων και οι αντίστοιχοι πίνακες κατακερματισμού ανάθεσης δώρων.



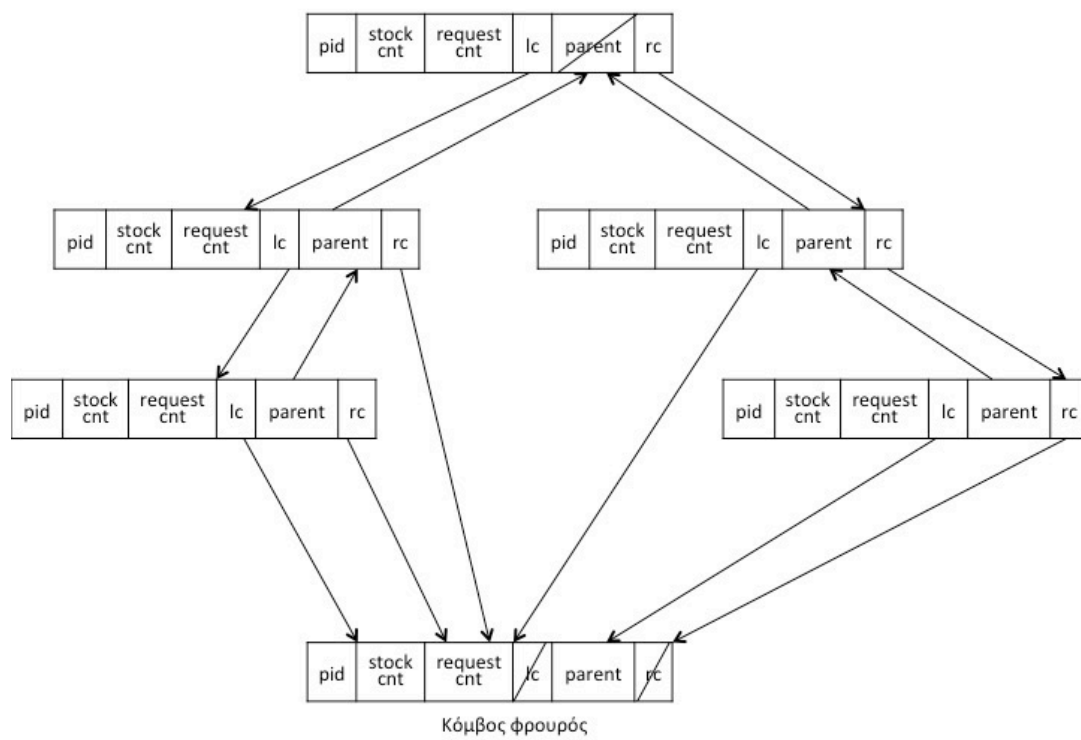
Σχήμα 1: Εισαγωγή κόμβων στο δέντρο παιδιών



Σχήμα 2: Πίνακας ηλικιακών κατηγοριών και τα αντίστοιχα δέντρα παιδιών για κάθε ηλικιακή κατηγορία

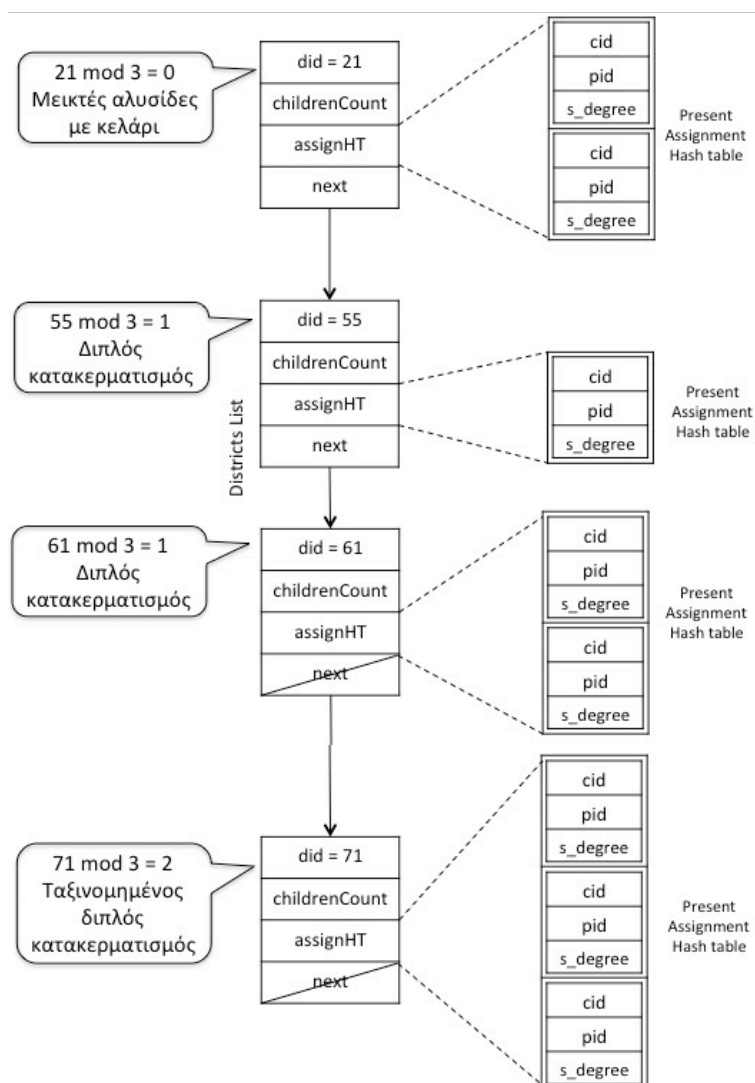


Σχήμα 3: απλά συνδεδεμένο, μη-ταξινομημένο δέντρο παιδιών μιας ηλικιακής κατηγορίας



Σχήμα 4: Διπλά –συνδεδεμένο, ταξινομημένο (ως προς το pid) δέντρο αποθεμάτων των δώρων με κόμβο φρουρό





Σχήμα 5: Ταξινομημένη (βάσει *did*), απλά-συνεδεμένη λίστα γεωγραφικών διαμερισμάτων και αντίστοιχοι πίνακες κατακερματισμού ανάθεσης δώρων για κάθε γεωγραφικό διαμέρισμα

**Τρόπος Λειτουργίας Προγράμματος**

Το πρόγραμμα που θα δημιουργηθεί θα πρέπει να εκτελείται καλώντας την ακόλουθη εντολή:

**<executable> <input-file>**

όπου <executable> είναι το όνομα του εκτελέσιμου αρχείου του προγράμματος (π.χ. a.out) και <input-file> είναι το όνομα ενός αρχείου εισόδου (π.χ. testfile) το οποίο περιέχει γεγονότα των ακόλουθων μορφών:

– **B <pid> <stock\_cnt>**

Γεγονός τύπου *buy present* το οποίο σηματοδοτεί την αγορά αποθέματος για το δώρο (present) με αναγνωριστικό <pid> από τον Άγιο Βασίλειο. Συγκεκριμένα, ο Άγιος Βασίλειος αγοράζει **απόθεμα** του δώρου με αναγνωριστικό <pid> ίσο με <stock\_cnt>. Μια εγγραφή για το δώρο για το οποίο αγοράστηκε απόθεμα θα πρέπει να εισάγεται στο δέντρο **αποθεμάτων των δώρων** στην κατάλληλη θέση ώστε το δέντρο να διατηρείται ταξινομημένο. Σημειώνεται ότι το ευριστικό δεν εφαρμόζεται κατά την εισαγωγή. Θεωρήστε ότι κάθε αναγνωριστικό δώρου που αναφέρεται από οποιοδήποτε γεγονός τύπου B ενός test-file θα είναι μοναδικό. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
B <pid> <stock_cnt>
    Presents = <pid1:stock1>, <pid2:stock2>, ..., <pidn:stockn>
DONE
```

όπου n είναι ο αριθμός των κόμβων στο δέντρο αποθεμάτων και για κάθε  $i \in \{1, \dots, n\}$ ,  $\text{pid}_i$  και  $\text{stock}_i$  είναι το αναγνωριστικό και το απόθεμα του δώρου που αντιστοιχεί στον i-οστό κόμβο του δέντρου όπως προκύπτει από την ενδοδιατεταγμένη διάσχισή του.

– **L <cid> <age> <did> <pid<sub>1</sub>> <pid<sub>2</sub>> <pid<sub>3</sub>>**

Γεγονός τύπου *letter received* το οποίο σηματοδοτεί την παραλαβή από τον Άγιο Βασίλειο ενός γράμματος που έχει αποσταλεί από ένα παιδί με αναγνωριστικό <cid>, ηλικίας <age> που διαμένει στην περιοχή με αναγνωριστικό <did>. Τα <pid<sub>1</sub>>, <pid<sub>2</sub>> και <pid<sub>3</sub>> είναι αναγνωριστικά δώρων που αντιστοιχούν στις προτιμήσεις του παιδιού. Για την εκτέλεση του γεγονότος αυτού απαιτείται να γίνουν οι ακόλουθες ενέργειες. Θα πρέπει να πραγματοποιηθεί πρόσβαση του κατάλληλου στοιχείου του πίνακα ηλικιακών κατηγοριών από το οποίο ξεκινάει το κατάλληλο δέντρο παιδιών όπου πρέπει να αποθηκευτεί η εγγραφή τύπου child για το παιδί (αν δεν υπάρχει ήδη εκεί). Αν το παιδί υπάρχει ήδη στο δέντρο, ενημερώνονται κατάλληλα τα πεδία του κόμβου που του αντιστοιχεί (θεωρήστε πως η ηλικία που παρέχεται στο test-file θα είναι σωστή αν το παιδί υπάρχει ήδη στο δέντρο).

Στη συνέχεια, θα πρέπει να γίνουν αναζητήσεις για τις προτιμήσεις του παιδιού στο δέντρο αποθεμάτων και για κάθε δώρο της προτίμησης του παιδιού θα πρέπει να αυξάνεται ο μετρητής ζήτησης (request\_cnt) κατά ένα. **Το συνολικό κόστος των αναζητήσεων πρέπει να είναι  $O(h)$  όπου h είναι το ύψος του δένδρου αποθεμάτων.** Είναι αξιοσημείωτο πως κατά την αναζήτηση εφαρμόζεται το ευριστικό που έχει περιγραφεί στην Ενότητα “Αναλυτική Περιγραφή της Ζητούμενης Υλοποίησης” παραπάνω.

Αν το δώρο με αναγνωριστικό <pid> δεν υπάρχει στη λίστα αποθεμάτων, ο Άγιος Βασίλειος θα πρέπει να προβαίνει στην αγορά 10 αποθεμάτων του δώρου αυτού. Σε αυτή την περίπτωση, ένας νέος κόμβος (που θα αντιστοιχεί σε αυτό το δώρο) θα πρέπει να εισάγεται στο δέντρο αποθεμάτων.

Επιπλέον, θα πρέπει να ελέγξετε αν το γεωγραφικό διαμέρισμα που διαμένει το παιδί υπάρχει ήδη στη λίστα γεωγραφικών διαμερισμάτων. Συνεπώς, όταν συναντάτε το πρώτο παιδί ενός γεωγραφικού διαμερίσματος θα πρέπει να δημιουργήσετε και να εισάγετε τον αντίστοιχο κόμβο στη λίστα γεωγραφικών διαμερισμάτων. *Σημείωση: αρχικά ο δείκτης προς τον πίνακα κατακερματισμού ανάθεσης δώρων θα είναι κενός (null). Ο πίνακας αυτός θα δημιουργηθεί κατά τα γεγονότα prepare presents όπως περιγράφεται παρακάτω. Αν το γεωγραφικό διαμέρισμα υπάρχει στη λίστα, θα πρέπει να αυξήσετε τον μετρητή των παιδιών που διαμένουν σ' αυτή την περιοχή (children\_cnt).* Αυτό χρειάζεται ώστε αργότερα να γνωρίζετε το μέγεθος του πίνακα κατακερματισμού που θα δεσμεύσετε για κάθε γεωγραφικό διαμέρισμα.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L <cid> <age> <did> <pid1> <pid2> <pid3>
```

```
DONE
```

## – P

Γεγονός τύπου *prepare presents* το οποίο σηματοδοτεί την ανάθεση των δώρων στα παιδιά λαμβάνοντας υπόψη τόσο τις προτιμήσεις τους όσο και το διαθέσιμο απόθεμα στην αποθήκη. Κατά το γεγονός αυτό, θα πρέπει να διατρέξετε τα δέντρα των παιδιών από όλες τις ηλικιακές κατηγορίες. Για κάθε παιδί θα πρέπει να εξετάζετε αν υπάρχει διαθέσιμο απόθεμα για κάθε ένα από τα δώρα της προτίμησής του με σειρά προτεραιότητας. Αν ο μετρητής αποθέματος (stock\_cnt) έχει τιμή μεγαλύτερη από το 0 τότε το δώρο ανατίθεται στο παιδί και ο μετρητής αποθέματος μειώνεται κατά ένα. Σε διαφορετική περίπτωση, θα πρέπει να εξετάζονται οι υπόλοιπες προτιμήσεις του παιδιού με τον ίδιο τρόπο. Αν καμιά από τις προτιμήσεις του παιδιού δεν μπορεί να ικανοποιηθεί με βάση τα αποθέματα της αποθήκης, τότε ανατίθεται στο παιδί ένα προκαθορισμένο δώρο με συγκεκριμένο αναγνωριστικό (pid = -2).

Στη συνέχεια θα πρέπει να εξετάσετε αν ο πίνακας κατακερματισμού ανάθεσης δώρων του γεωγραφικού διαμερίσματος δεν έχει δημιουργηθεί (έχει τιμή null). Αυτό θα συμβαίνει κατά την πρώτη ανάθεση δώρου σε κάποιο παιδί αυτού του γεωγραφικού διαμερίσματος. Σ' αυτήν την περίπτωση θα δημιουργείτε τον πίνακα κατακερματισμού του γεωγραφικού διαμερίσματος με μέγεθος ίσο με τον αριθμό των παιδιών που διαμένουν σ' αυτό (σημειώνεται ότι το μέγεθος καθορίζεται από το πεδίο children\_cnt της εγγραφής τύπου district που αντιστοιχεί σε αυτό το γεωγραφικό διαμέρισμα).

Τέλος, θα πραγματοποιείται την εισαγωγή στον πίνακα κατακερματισμού ανάθεσης δώρων του γεωγραφικού διαμερίσματος που διαμένει το παιδί. Για να καθορίσετε τη μέθοδο που θα εφαρμόσετε στον πίνακα κατακερματισμού θα εφαρμόσετε την τεχνική που έχει περιγραφεί στην Ενότητα “Αναλυτική Περιγραφή της Ζητούμενης Υλοποίησης” παραπάνω.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
P
DISTRICTS:
  <did1>
    ASSIGNMENT: <cid1,1, pid1,1>, ..., <cid1,n1, pid1,n1>
  <did2>
    ASSIGNMENT: <cid2,1, pid2,1>, ..., <cid2,n2, pid2,n2>
  ...
  <didk>
    ASSIGNMENT: <cidk,1, pidk,1>, ..., <cidk,nk, pidk,nk>
DONE
```

όπου  $k$  είναι ο αριθμός των γεωγραφικών διαμερισμάτων στη λίστα γεωγραφικών διαμερισμάτων, για κάθε  $j$ ,  $1 \leq j \leq k$ ,  $n_j$  είναι το μέγεθος του πίνακα κατακερματισμού ανάθεσης δώρων του  $j$ -οστού γεωγραφικού διαμερίσματος, και για κάθε  $i \in \{1, \dots, n_j\}$ ,  $\langle \text{pid}_{k,i}, \text{cid}_{k,i} \rangle$  είναι το αναγνωριστικό του παιδιού και του δώρου του  $i$ -οστού κελιού στον πίνακα κατακερματισμού ανάθεσης δώρων του γεωγραφικού διαμερίσματος με αναγνωριστικό  $k$ .

#### – F <cid> <did> <s\_degree>

Γεγονός τύπου *give feedback* το οποίο σηματοδοτεί την εισαγωγή κάποιου βαθμού <s\_degree> ικανοποίησης του παιδιού με αναγνωριστικό <cid> για το δώρο που έλαβε. Η τιμή του s\_degree είναι ένας ακέραιος αριθμός από το 1 ως το 5, με το 1 να αντιστοιχεί στον ελάχιστο βαθμό ικανοποίησης του παιδιού και το 5 στο μέγιστο. Κατά το γεγονός αυτό, εντοπίζεται το παιδί με αναγνωριστικό <cid> στον πίνακα κατακερματισμού του γεωγραφικού διαμερίσματος που διαμένει το παιδί και τίθεται ο βαθμός ικανοποίησής του ώστε να είναι ίσος με <s\_degree>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
F <cid> <s_degree> <pid>
DONE
```

όπου <pid> είναι το αναγνωριστικό του δώρου που έλαβε το παιδί με αναγνωριστικό <cid> και έδωσε βαθμό ικανοποίησης <s\_degree>.

#### – N

Γεγονός τύπου *new season* το οποίο σηματοδοτεί την έναρξη νέας περιόδου προετοιμασίας του Αγίου Βασιλείου. Για κάθε παιδί κάθε ηλικιακής κατηγορίας, αυξάνουμε την ηλικία του κατά ένα. Αυτό ενδέχεται να προκαλέσει τη μετακίνηση κάποιων παιδιών σε νέα ηλικιακή κατηγορία. Ο αλγόριθμος μετακίνησης των παιδιών κάθε ηλικιακής κατηγορίας στην επόμενη πρέπει να γίνεται σε χρόνο,

$O(n*m)$ , όπου  $n$  είναι ο αριθμός των κόμβων στο δένδρο παιδιών που αντιστοιχεί στη μία ηλικιακή κατηγορία και  $m$  είναι ο αριθμός των κόμβων στο δένδρο που αντιστοιχεί στην άλλη ηλικιακή κατηγορία.

Τα παιδιά που υπερβαίνουν το 15ο έτος της ηλικίας τους θα πρέπει να αφαιρεθούν από τη δομή. Η διαγραφή θα πραγματοποιείται με τη μέθοδο που έχει περιγραφεί στην Ενότητα “Αναλυτική Περιγραφή της Ζητούμενης Υλοποίησης” παραπάνω. Επιπρόσθετα, κάθε ένα από τα στοιχεία του πίνακα προτιμήσεων (`present_choices`) του παιδιού πρέπει να αρχικοποιηθεί με την τιμή  $-1$ , που υποδηλώνει πως το παιδί δεν έχει στείλει γράμμα στον Άγιο Βασίλειο για το τρέχον έτος και άρα δεν θα πρέπει να του αποδοθεί δώρο αν δεν ληφθεί τέτοιο γράμμα μέχρι το επόμενο event τύπου `P`. Επιπλέον, για κάθε γεωγραφικό διαμέρισμα (`district`) της λίστας γεωγραφικών διαμερισμάτων, θα πρέπει να αποδεσμευτεί ο πίνακας κατακερματισμού ανάθεσης δώρων και να μηδενιστεί ο μετρητής των παιδιών που ανήκουν στο γεωγραφικό διαμέρισμα. Επίσης, θα πρέπει να διατρέξετε το δέντρο αποθεμάτων των δώρων μια φορά και να πραγματοποιήσετε τις ακόλουθες ενέργειες: (α) να διαγράψετε τις εγγραφές που αντιστοιχούν σε εκείνα τα δώρα για τα οποία ο μετρητής αποθεμάτων είναι 0 και (β) να μηδενίσετε το μετρητή ζήτησης των υπολοίπων.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

N

DONE

– C

Γεγονός τύπου *clear list of children* το οποίο σηματοδοτεί τη διαγραφή των παιδιών (`child`) τα οποία δεν έχουν στείλει γράμμα στον Άγιο Βασίλειο από την τελευταία αλλαγή της σεζόν. Συγκεκριμένα, για κάθε παιδί κάθε ηλικιακής κατηγορίας, θα πρέπει να γίνεται έλεγχος στον πίνακα των προτιμήσεων του (`present_choices`) και αν όλες οι θέσεις έχουν την τιμή  $-1$ , τότε το παιδί θα πρέπει να διαγράφεται από τη δομή αφού αυτό σημαίνει ότι δεν έχει στείλει στο τρέχον έτος γράμμα στον Άγιο Βασίλειο. Η διαγραφή θα εκτελείται με τη μέθοδο που έχει περιγραφεί στην Ενότητα “Αναλυτική Περιγραφή της Ζητούμενης Υλοποίησης” παραπάνω. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

C

0-3: <cid<sub>1,1</sub>>, ..., <cid<sub>1,n1</sub>>

4-7: <cid<sub>2,1</sub>>, ..., <cid<sub>2,n2</sub>>

8-11: <cid<sub>3,1</sub>>, ..., <cid<sub>3,n3</sub>>

12-15: <cid<sub>4,1</sub>>, ..., <cid<sub>4,n4</sub>>

DONE

όπου για κάθε  $k \in \{1, \dots, 4\}$ ,  $n_k$  είναι ο αριθμός των κόμβων στο δέντρο παιδιών της ηλικιακής κατηγορίας το οποίο δεικτοδοτείται από την  $k$ -οστή θέση του πίνακα *Age\_categories* και για κάθε  $i \in \{1, \dots, n\}$ ,  $\langle cid_{k,i} \rangle$  είναι το αναγνωριστικό του παιδιού που αντιστοιχεί στον  $i$ -οστό κόμβο του δέντρου παιδιών της ηλικιακής κατηγορίας αυτής κατά την ενδοδιατεταγμένη διάσχιση.

#### – S <pid>

Γεγονός τύπου *search present* το οποίο σηματοδοτεί την αναζήτηση του δώρου με αναγνωριστικό <pid> στο δέντρο αποθεμάτων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
S <pid>
    <stock_cnt>, <request_cnt>
DONE
```

όπου <stock\_cnt> και <request\_cnt> είναι ο μετρητής αποθέματος και ο μετρητής ζήτησης του δώρου με αναγνωριστικό <pid>, αντίστοιχα. Αν δεν υπάρχει δώρο με αναγνωριστικό <cid> στο δέντρο, θα πρέπει να εμφανίζεται και στους δυο μετρητές η τιμή 0.

#### – H

Γεγονός τύπου *print children* το οποίο σηματοδοτεί την εκτύπωση του δέντρου παιδιών όλων των ηλικιακών κατηγοριών. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
H
0-3:  <cid1,1:{choice1, choice2, choice3}>, ..., <cid1,n1:{choice1, choice2, choice3}>
4-7:  <cid2,1:{choice1, choice2, choice3}>, ..., <cid2,n2:{choice1, choice2, choice3}>
8-11: <cid3,1:{choice1, choice2, choice3}>, ..., <cid3,n3:{choice1, choice2, choice3}>
12-15: <cid4,1:{choice1, choice2, choice3}>, ..., <cid4,n4:{choice1, choice2, choice3}>
DONE
```

όπου για κάθε  $k \in \{1, \dots, 4\}$ ,  $n_k$  είναι ο αριθμός των κόμβων στο δέντρο παιδιών της ηλικιακής κατηγορίας η οποία δεικτοδοτείται από την  $k$ -οστή θέση του πίνακα *Age\_categories* και για κάθε  $i \in \{1, \dots, n\}$ ,  $cid_{k,i}$  είναι το αναγνωριστικό του παιδιού που αντιστοιχεί στον  $i$ -οστό κόμβο του δέντρου παιδιών της ηλικιακής κατηγορίας αυτής εφαρμόζοντας ενδοδιατεταγμένη διάσχιση.

#### – I

Γεγονός τύπου *print districts* το οποίο σηματοδοτεί την εκτύπωση των αναγνωριστικών των παιδιών και των αναγνωριστικών των δώρων που έχουν λάβει από τον Άγιο Βασίλειο που κατοικούν σε όλα τα

γεωγραφικά διαμερίσματα. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
I
  DISTRICTS:
    <did1>
      ASSIGNMENTS: <cid1,1:pid1,1:feedback1,1> ... <cid1,n1:pid1,n1:feedback1,n1>
    <did2>
      ASSIGNMENTS: <cid2,1:pid2,1:feedback2,1> ... <cid2,n2:pid2,n2:feedback2,n2>
    ...
    <didk>
      ASSIGNMENTS: <cidk,1:pidk,1:feedbackk,1> ... <cidk,nk:pidk,nk:feedbackk,nk>
  DONE
```

όπου  $k$  είναι ο αριθμός των γεωγραφικών διαμερισμάτων στη λίστα γεωγραφικών διαμερισμάτων, για κάθε  $j$ ,  $1 \leq j \leq k$ ,  $n_j$  είναι το μέγεθος του πίνακα κατακερματισμού ανάθεσης δώρων του  $j$ -οστού γεωγραφικού διαμερίσματος, και για κάθε  $i \in \{1, \dots, n_j\}$

– **T**

Γεγονός τύπου *print stock* το οποίο σηματοδοτεί την εκτύπωση των αναγνωριστικών όλων των δώρων (είτε υπάρχει απόθεμα, είτε όχι) στο δέντρο αποθεμάτων των δώρων. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
T
  Stock = <pid1:stock1>, ..., <pidn:stockn>
  DONE
```

όπου  $n$  είναι ο αριθμός των κόμβων στο δέντρο αποθεματικών και για κάθε  $i \in \{1, \dots, n\}$ ,  $pid_i$  και  $stock_i$  είναι το αναγνωριστικό και το απόθεμα του δώρου που αντιστοιχεί στο  $i$ -οστό κόμβο του δέντρου αποθεμάτων.

[Bonus 15%]

– **A <k>**

Γεγονός τύπου *analytics* κατά το οποίο θα εντοπιστούν τα  $\langle k \rangle$  (δίνεται ως παράμετρος) πιο δημοφιλή δώρα βάσει του πεδίου *request\_cnt*. Για το σκοπό αυτό θα χρησιμοποιήσετε έναν βοηθητικό πίνακα  $\langle k \rangle$  θέσεων στον οποίο θα αποθηκεύονται δείκτες προς τους κόμβους του δέντρου αποθεμάτων.

Πιο συγκεκριμένα θα εφαρμόσετε τις εξής ενέργειες:

- Ο βοηθητικός πίνακας αρχικοποιείται με τα  $\langle k \rangle$  πρώτα δώρα του δέντρου αποθεμάτων (σύμφωνα με την ενδοδιατεταγμένη διάσχιση του δένδρου). Ο πίνακας αυτός θα πρέπει να υλοποιεί ουρά προτεραιότητας, στην οποία η προτεραιότητα καθορίζεται από τη ζήτηση του κάθε δώρου (ζητείται

δηλαδή να υλοποιηθεί σωρός). Για να μετατρέψετε τον πίνακα σε σωρό μετά την αρχικοποίηση του θα πρέπει να εκτελείτε την `Heapify()` όπως έχει διδαχθεί στο μάθημα.

- ii. Πραγματοποιείται ενδοδιατεταγμένη διάσχιση του δένδρου αποθεμάτων και για κάθε δώρο, εξετάζεται αν ο μετρητής ζήτησης του είναι μεγαλύτερος από τον αντίστοιχο μετρητή του δώρου με τη μικρότερη ζήτηση στον βοηθητικό πίνακα. Αν συμβαίνει αυτό τότε αντικαθιστούμε το δώρο με τη μικρότερη ζήτηση στον πίνακα με το τρέχον δώρο από το δέντρο αποθεμάτων. Αυτό θα πραγματοποιείται καλώντας πρώτα την `HeapDeleteMin()`, η οποία θα διαγράφει το στοιχείο με την μικρότερη προτεραιότητα στο σωρό και στη συνέχεια την `HeapInsert()` με παράμετρο το νέο κλειδί που πρέπει να εισαχθεί. Με αυτό τον τρόπο, ο βοηθητικός πίνακας εξακολουθεί να είναι σωρός.
- iii. Μετά το τέλος της διάσχισης του δένδρου αποθεμάτων, ο βοηθητικός πίνακας θα περιέχει δείκτες στους κόμβους του δένδρου που αντιστοιχούν στα  $<k>$  πιο δημοφιλή δώρα.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
A
    <pidi> : <request_cnti>
    ...
    <pidk> : <request_cntk>
DONE
```

για κάθε  $i \in \{1, \dots, k\}$ ,  $<pid_i>$  είναι το αναγνωριστικό του  $i$ -οστού πιο δημοφιλές δώρου στο δέντρο αποθεμάτων και  $<request\_cnt_i>$  είναι ο μετρητής ζήτησης του δώρου αυτού.



**Βαθμολογία**

B	Buy present	10
L	Letter received	15
P	Prepare presents	15
F	Give feedback	10
N	New season	15
C	Clear list of children	10
S	Search present	5
H	Print child	3
I	Print district	3
T	Print stock	4
A	[Bonus] Analytics	15
Δεν κάνει compile		5
Δεν τρέχει και δεν τρέχουν τα test-files		5

**Δομές Δεδομένων**

Στην υλοποίησή σας δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες δομές δεδομένων (πχ., ArrayList) είτε η υλοποίηση πραγματοποιηθεί στη C είτε στη Java. Στη συνέχεια παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας.

```
#define M 3
#define N 4

typedef struct child {
    int cid;
    int age;
    int did;
    int present_choices[M];
    struct child *lc;
    struct child *rc;
} Child;

typedef struct present {
    int pid;
    int stock_cnt;
    int request_cnt;
    struct present *parent;
    struct present *lc;
    struct present *rc;
} Present;

typedef struct present_assign {
    int cid;
    int pid;
    int s_degree;
} PresentAssign;

typedef struct district {
    int did;
    int children_cnt;
```

```
    PresentAssign *assignHT; /* pointer to hashTable*/
    struct district *next; /* Singly-linked, sorted by did */
} District;

Child *Age_categories[N];

/* global variable pointing to the root of stock tree */
Present *stock_R;

/*global variable pointing to the sentinel node of stock tree */
Present *stock_S;
```