

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2014-15
Διδάσκουσα: Παναγιώτα Φατούρου
Προγραμματιστική Εργασία - 2^ο Μέρος

Ημερομηνία Παράδοσης: Παρασκευή, 19 Δεκεμβρίου 2014, ώρα 23:59.

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

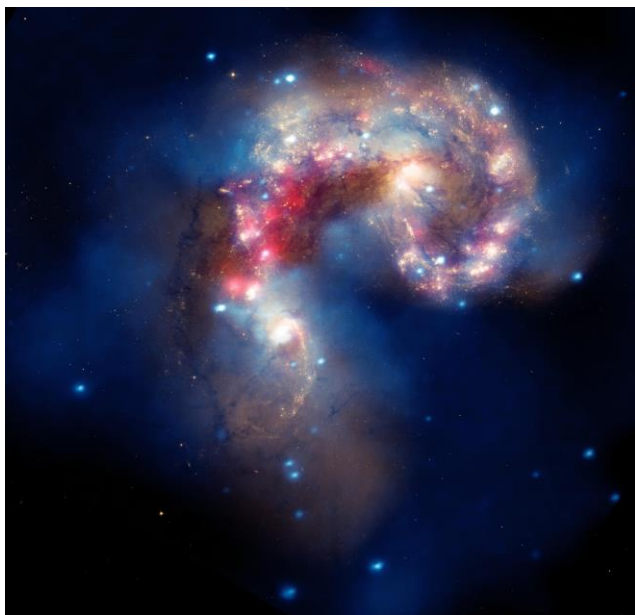


Image credit: NASA/CXC/SAO/JPL-Caltech/STScI



Image credit: NASA/CXC/JPL-Caltech/STScI

Γενική Περιγραφή

Κατά αντιστοιχία με το 1^ο μέρος της εργασίας σας, καλείστε να υλοποιήσετε ένα πρόγραμμα που προσομοιώνει ένα μέρος του σύμπαντος. Το σύμπαν περιέχει γαλαξίες. Κάθε γαλαξίας περιέχει ηλιακά συστήματα και κάποιους ορφανούς πλανήτες (κι όχι συστάδες πλανητών όπως συνέβαινε στο 1^ο μέρος της εργασίας σας). Κάθε ηλιακό σύστημα έχει ένα αστέρι-ήλιο γύρο από το οποίο περιστρέφονται πλανήτες.

Όπως και στο 1^ο μέρος αυτής της προγραμματιστικής άσκησης, όλα θα ξεκινήσουν (αρχικοποιηθούν) με μία Μεγάλη Έκρηξη. Η «Μεγάλη Έκρηξη» σηματοδοτεί την έναρξη της δημιουργίας των γαλαξιών, των ηλιακών συστημάτων, των πλανητών και των υπόλοιπων σωμάτων που υπάρχουν στο σύμπαν.

Σημείωση

Σημειώνεται ότι προκειμένου να πραγματοποιηθούν οι εκπαιδευτικοί στόχοι της εργασίας, όσα αναγράφονται στην παρούσα εκφώνηση δεν αντιστοιχούν (απαραίτητα) στην επιστημονική θεωρία δημιουργίας και εξέλιξης του σύμπαντος με ακρίβεια ή σε ό,τι θεωρείται αποδεκτό από την επιστημονική κοινότητα για τη δημιουργία του σύμπαντος και των μερών που το αποτελούν.

Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

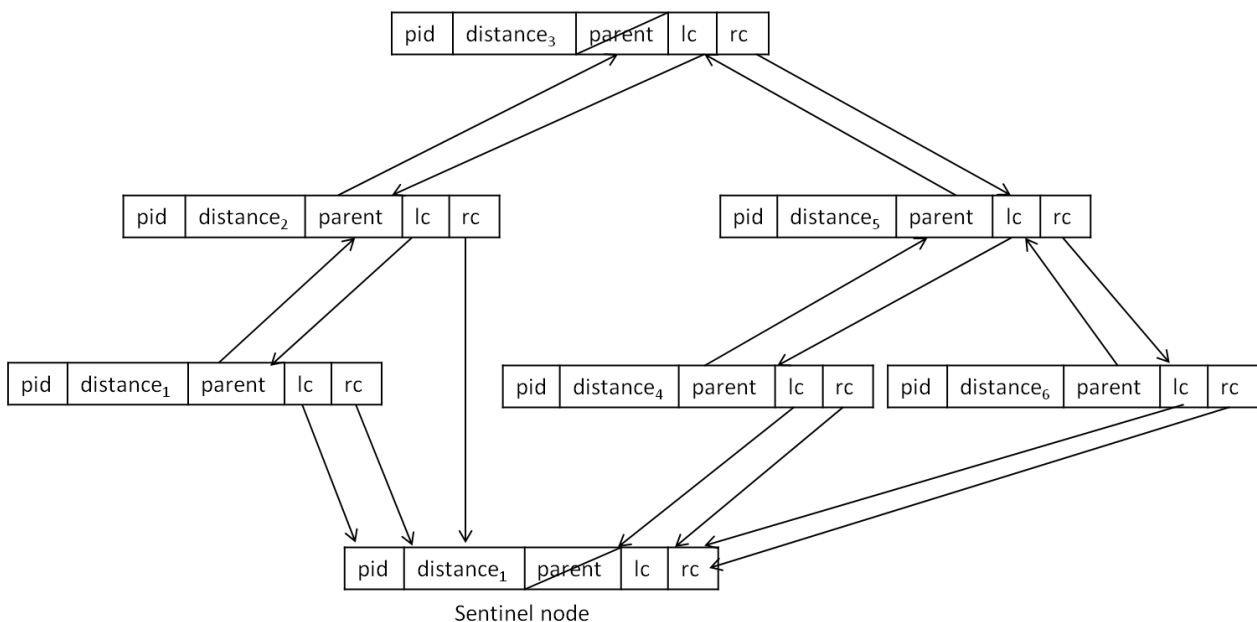
Για την υλοποίηση της προσομοίωσης της δημιουργίας του σύμπαντος, θα πρέπει να χρησιμοποιήσετε τον ίδιο πίνακα γαλαξιών (*Galaxies*), όπως και στο 1^ο μέρος της εργασίας. Ο πίνακας αυτός έχει σταθερό μέγεθος *N*. Κάθε στοιχείο του *πίνακα γαλαξιών* είναι μια εγγραφή (ένα *struct*) τύπου *galaxy_t* με τα ακόλουθα πεδία:

- **gid:** Αναγνωριστικό (τύπου *int*) του γαλαξία. Ένας μοναδικός αριθμός που χαρακτηρίζει μοναδικά το γαλαξία.
- **solars:** Δείκτης (τύπου *solar_t**) στο πρώτο στοιχείο μιας **μη-ταξινομημένης, απλά-συνδεδεμένης** λίστας με **κόμβο φρουρό**, κάθε στοιχείο της οποίας αντιστοιχεί σε ένα ηλιακό σύστημα που ανήκει στο γαλαξία με αναγνωριστικό *gid*. Η λίστα αυτή ονομάζεται *λίστα ηλιακών συστημάτων* του γαλαξία με αναγνωριστικό *gid*. Κάθε στοιχείο της *λίστας ηλιακών συστημάτων* ενός γαλαξία είναι μία εγγραφή (ένα *struct*) τύπου *solar_t* με τα ακόλουθα πεδία:
 - **sid:** Αναγνωριστικό (τύπου *int*) του αστεριού-ήλιου του ηλιακού συστήματος. Είναι ένας μοναδικός αριθμός που αντιστοιχεί στο αστέρι-ήλιο του ηλιακού συστήματος.
 - **planets:** Ένας δείκτης (τύπου *planet_t**) στη ρίζα ενός **διπλά-συνδεδεμένου δυαδικού δένδρου (binary search tree)** με **κόμβο φρουρό**, το οποίο είναι **ταξινομημένο ως προς την απόσταση του κάθε πλανήτη από το αστέρι-ήλιο** του ηλιακού συστήματος. Ένα δένδρο πλανητών φαίνεται στο Σχήμα 1. Κάθε στοιχείο του *δέντρου πλανητών* ενός ηλιακού συστήματος αντιστοιχεί σε έναν πλανήτη του ηλιακού συστήματος αποτελώντας μία εγγραφή (ένα *struct*) τύπου *planet_t* με τα ακόλουθα πεδία:
 - **pid:** Αναγνωριστικό (τύπου *int*) του πλανήτη. Ένας μοναδικός αριθμός που αντιστοιχεί στον πλανήτη.
 - **distance:** Ένας αριθμός που αντιστοιχεί στην απόσταση του πλανήτη με αναγνωριστικό *pid* **από το αστέρι/ήλιο του ηλιακού συστήματος στο οποίο ανήκει**. Προσέξτε πως το πεδίο αυτό αποθηκεύει διαφορετική πληροφορία από ότι στο 1^ο μέρος της προγραμματιστικής εργασίας, όπου αποθηκευόταν η απόσταση του πλανήτη από τον αμέσως προηγούμενο πλανήτη βάσει της σειράς εγγύτητας στο αστέρι-ήλιος.
 - **parent:** Δείκτης (τύπου *planet_t**) στον πατρικό κόμβο του κόμβου που αντιστοιχεί στον πλανήτη με αναγνωριστικό *pid* στο δένδρο πλανητών
 - **lc:** Δείκτης (τύπου *planet_t**) στο αριστερό παιδί του κόμβου που αντιστοιχεί στον πλανήτη με αναγνωριστικό *pid* στο δένδρο πλανητών
 - **rc:** Δείκτης (τύπου *planet_t**) στο δεξί παιδί του κόμβου που αντιστοιχεί στον πλανήτη με αναγνωριστικό *pid* στο δένδρο πλανητών
 - **psentinel:** Δείκτης (τύπου *planet_t**) στον κόμβο φρουρό του δέντρου πλανητών
 - **next:** Δείκτης (τύπου *solar_t**) στον επόμενο κόμβο της *λίστας ηλιακών συστημάτων* του γαλαξία.
- **ssentinel:** Δείκτης (τύπου *solar_t**) στον κόμβο φρουρό της *λίστας ηλιακών συστημάτων* του γαλαξία με αναγνωριστικό *gid*. Καθώς το πεδίο *next* του κόμβου

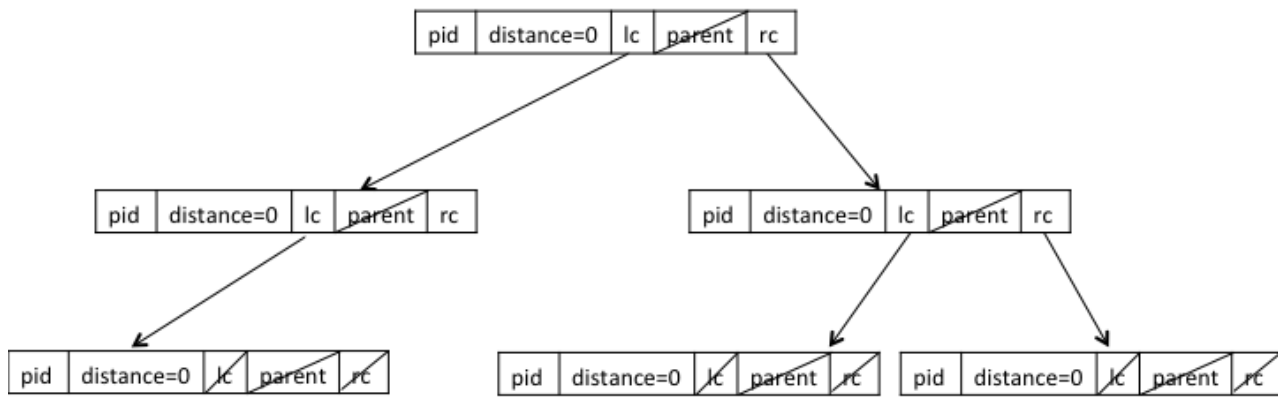
φρουρού δεν χρειάζεται για την υλοποίηση της λίστας, κατ' εξαίρεση σε αυτή την εργασία θα το χρησιμοποιήσουμε ως δείκτη στον τελευταίο κόμβο της λίστας. Με αυτό τον τρόπο μπορούμε να επιτύχουμε την συνένωση δύο *λιστών ηλιακών συστημάτων* σε χρόνο $O(1)$.

- **orphans:** Ένας δείκτης (τύπου `planet_t*`) στη ρίζα ενός **απλά-συνδεδεμένου δυαδικού δένδρου**. Το δέντρο αυτό ονομάζεται *δένδρο ορφανών πλανητών* του γαλαξία με αναγνωριστικό `gid`. Κάθε κόμβος του δένδρου είναι μία εγγραφή (`struct`) τύπου `planet_t` όπως έχει περιγραφεί παραπάνω. Στο δένδρο ορφανών πλανητών, οι τιμές του πεδίου `distance` της εγγραφής τύπου `planet_t` δεν έχουν ενδιαφέρον (επομένως περιέχουν απλά μια αρχική τιμή). Επίσης, οι δείκτες `parent` όλων των κόμβων του δένδρου είναι `NULL`. Το δένδρο ορφανών πλανητών δεν είναι ταξινομημένο (δηλαδή το δένδρο δεν είναι **binary search tree**). Εισαγωγές στο δένδρο αυτό θα πραγματοποιούνται όπως περιγράφονται στην Άσκηση 1 α) της 3^{ης} σειράς ασκήσεων. Ένα δένδρο ορφανών πλανητών φαίνεται στο Σχήμα 2. Προσέξτε πως στο μέρος αυτό της προγραμματιστικής εργασίας δεν υπάρχουν πλέον συστάδες πλανητών. Κάθε ορφανός πλανήτης κινείται (χωρίς να ανήκει σε κάποια συστάδα) μέσα στο γαλαξία στον οποίο ανήκει.

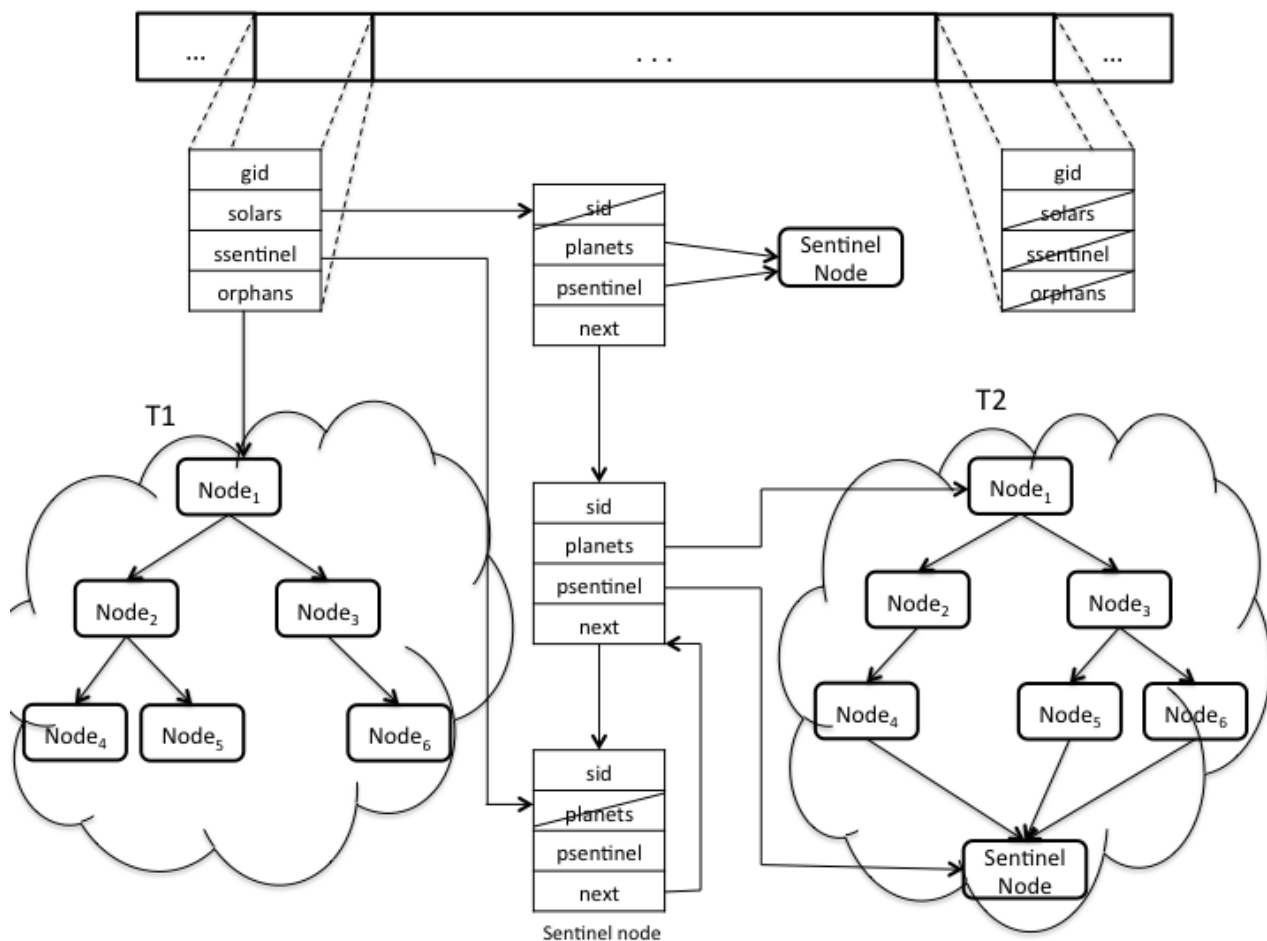
Το Σχήμα 3 παρουσιάζει τις δομές δεδομένων που περιγράφηκαν παραπάνω σε σχηματική μορφή.



Σχήμα 1: Μορφή δένδρου πλανητών.



Σχήμα 2: Μορφή δέντρου ορφανών πλανητών.



Σχήμα 3: Σχηματική μορφή δομών δεδομένων που ζητείται να υλοποιηθούν στο 2^ο μέρος της προγραμματιστικής εργασίας. Τα δένδρα T1 και T2 παρουσιάζονται πιο αναλυτικά στα Σχήματα 1 και 2, αντίστροφα.

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:** Γεγονός τύπου *Big Bang* το οποίο σηματοδοτεί τη δημιουργία του σύμπαντος. Κατά το γεγονός αυτό αρχικοποιείται η δομή του **πίνακα γαλαξιών** (Galaxies) όπου το κάθε πεδίο του κάθε κελιού του πίνακα αρχικοποιείται με την τιμή NULL αν είναι δείκτης και INT_MAX (ή Integer.MAX_VALUE) για υλοποίηση σε γλώσσα C (ή Java αντίστοιχα) αν είναι τύπου `int`. Το γεγονός αυτό υλοποιείται με τον ίδιο τρόπο όπως και στο πρώτο μέρος της προγραμματιστικής εργασίας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
B DONE
```

- **G <gid>:** Γεγονός τύπου *Galaxy Creation* το οποίο σηματοδοτεί τη δημιουργία ενός νέου γαλαξία (galaxy_t) με αναγνωριστικό <gid> στο σύμπαν. Ο νέος γαλαξίας περιέχει κενή **λίστα ηλιακών συστημάτων** (solars) (δηλαδή η λίστα ηλιακών συστημάτων του γαλαξία περιέχει μόνο τον κόμβο φρουρό) και κενό **δένδρο ορφανών πλανητών** (orphans). Ο νέος γαλαξίας προστίθεται στον **πίνακα γαλαξιών**. Χρησιμοποιήστε κατάλληλες μεταβλητές ώστε η χρονική πολυπλοκότητα της εισαγωγής να είναι O(1). Το γεγονός αυτό υλοποιείται με τον ίδιο τρόπο όπως και στο πρώτο μέρος της προγραμματιστικής εργασίας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
G <gid>
  Galaxies = <gid1>, <gid2>, ... <gidn>
DONE
```

όπου n είναι το πλήθος των γαλαξιών που υπάρχουν στο σύμπαν και για κάθε i , $1 \leq i \leq n$, **<gid_i>** είναι το αναγνωριστικό του i -οστού γαλαξία στο σύμπαν.

- **S <sid> <gid>:** Γεγονός τύπου *Star Birth* το οποίο σηματοδοτεί τη δημιουργία ενός νέου αστεριού-ήλιου και κατά συνέπεια τη δημιουργία ενός νέου ηλιακού συστήματος (solar_t) με αναγνωριστικό αστεριού-ήλιου <sid> στο γαλαξία με αναγνωριστικό <gid>. Το νέο ηλιακό σύστημα περιέχει κενό **δέντρο πλανητών** (planets), δηλαδή το δένδρο πλανητών του ηλιακού συστήματος περιέχει μόνο τον κόμβο φρουρό. Το γεγονός αυτό προσθέτει ένα νέο στοιχείο (που αντιστοιχεί στο νέο ηλιακό σύστημα) στη **λίστα ηλιακών συστημάτων** (solars) του γαλαξία με αναγνωριστικό <gid>. Το γεγονός αυτό υλοποιείται με τον ίδιο τρόπο όπως και στο πρώτο μέρος της προγραμματιστικής εργασίας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
S <sid> <gid>
  Solars = <sid1>, <sid2> ... <sidn>
DONE
```

όπου n είναι το πλήθος των ηλιακών συστημάτων που υπάρχουν στο γαλαξία με αναγνωριστικό $\langle \text{gid} \rangle$ και για κάθε i , $1 \leq i \leq n$, $\langle \text{sid}_i \rangle$ είναι το αναγνωριστικό του i -οστού ηλιακού συστήματος στο γαλαξία.

- **P $\langle \text{pid} \rangle$ $\langle \text{distance} \rangle$ $\langle \text{sid} \rangle$:** Γεγονός τύπου *Planet Creation* το οποίο σηματοδοτεί τη δημιουργία ενός νέου πλανήτη (`planet_t`) με αναγνωριστικό $\langle \text{pid} \rangle$ στο ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid} \rangle$. Η παράμετρος $\langle \text{distance} \rangle$ περιγράφει την απόσταση του νέου πλανήτη από το αστέρι-ήλιο του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid} \rangle$. Το γεγονός αυτό προσθέτει το νέο πλανήτη στην κατάλληλη θέση, βάσει του $\langle \text{distance} \rangle$, στο *δέντρο πλανητών* του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid} \rangle$. **Είναι αξιοσημείωτο ότι** για αυτή τη φάση της εργασίας, το πεδίο `distance` που αποθηκεύεται στην εγγραφή, αντιστοιχεί στην απόλυτη απόσταση του πλανήτη από το αστέρι/ήλιο του ηλιακού συστήματος που ανήκει. Προσέξτε πως το γεγονός αυτό απαιτεί να βρεθεί ο γαλαξίας στον οποίο ανήκει το ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid} \rangle$. Μετά το πέρας της εκτέλεσης ενός γεγονότος τύπου P το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
P <pid> <distance> <sid>
  <pid1> : <distance1>
  <pid2> : <distance2>
  ...
  <pidn> : <distancen>
DONE
```

όπου n είναι το πλήθος των πλανητών που υπάρχουν στο ηλιακό σύστημα με αναγνωριστικό αστεριού-ηλίου $\langle \text{sid} \rangle$ και για κάθε i , $1 \leq i \leq n$:

- $\langle \text{pid}_i \rangle$ είναι το αναγνωριστικό του i -οστού πλανήτη στο δέντρο πλανητών του ηλιακού συστήματος με αναγνωριστικό αστεριού-ηλίου $\langle \text{sid} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση),
- $\langle \text{distance}_i \rangle$ είναι η απόσταση του πλανήτη με αναγνωριστικό $\langle \text{pid}_i \rangle$ από το αστέρι/ήλιο του ηλιακού συστήματος με αναγνωριστικό αστεριού-ηλίου $\langle \text{sid} \rangle$,

Προσοχή: Για την εκτύπωση των αναγνωριστικών των πλανητών ενός ηλιακού συστήματος θα πρέπει να εφαρμόζεται *ενδοδιατεταγμένη διάσχιση* στο δένδρο πλανητών.

- **D $\langle \text{sid} \rangle$ $\langle \text{distance} \rangle$:** Γεγονός τύπου *Star Death* το οποίο σηματοδοτεί το θάνατο ενός αστεριού-ηλίου και κατά συνέπεια την καταστροφή του ηλιακού συστήματος με αναγνωριστικό αστεριού-ηλίου $\langle \text{sid} \rangle$. Κατά την καταστροφή του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid} \rangle$ καταστρέφονται (και άρα πρέπει να διαγραφούν) και όλοι οι πλανήτες του που απέχουν λιγότερο από $\langle \text{distance} \rangle$ km από το αστέρι-ήλιο. Οι πλανήτες σε μεγαλύτερη απόσταση μετατρέπονται σε ορφανούς πλανήτες και κατά συνέπεια εισάγονται στο δέντρο ορφανών πλανητών. Το συνολικό κόστος για την μεταφορά όλων των πλανητών που μετατρέπονται σε ορφανούς πλανήτες στο δένδρο

ορφανών πλανητών θα πρέπει να είναι $O(n)$, όπου n είναι **το μέγιστο** μεταξύ των στοιχείων του δένδρου ορφανών πλανητών του γαλαξία στον οποίο ανήκει το αστέρι-ήλιος με αναγνωριστικό `<sid>` και του δένδρου πλανητών του ηλιακού συστήματος του οποίου το αστέρι-ήλιος έχει αναγνωριστικό `<sid>`. *Σημειώνουμε ότι, το πεδίο `distance` της εγγραφής στο νέο δέντρο θα πρέπει να έχει την τιμή 0.* Τέλος, το κατεστραμμένο ηλιακό σύστημα θα πρέπει να διαγράφεται από τη **λίστα ηλιακών συστημάτων** του γαλαξία όπου ανήκει. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
D <sid> <distance>
  Solars    = <sid1>, <sid2>, ... <sidn>
  Orphans   = <oid1>, <oid2>, ... <oidk>
DONE
```

όπου n είναι το πλήθος των ηλιακών συστημάτων που υπάρχουν στο γαλαξία στον οποίο ανήκει το προς διαγραφή ηλιακό σύστημα, μετά τη διαγραφή του ηλιακού συστήματος με αναγνωριστικό αστεριού-ηλίου `<sid>`, k είναι το πλήθος των ορφανών πλανητών στο **δέντρο ορφανών πλανητών** του γαλαξία που ανήκει το προς διαγραφή ηλιακό σύστημα και:

- για κάθε i , $1 \leq i \leq n$, `<sidi>` είναι το αναγνωριστικό του i -οστού ηλιακού συστήματος στο γαλαξία όπου ανήκει το προς διαγραφή ηλιακό σύστημα,
- για κάθε i , $1 \leq i \leq k$, `<oidi>` είναι το αναγνωριστικό του i -οστού ορφανού πλανήτη στο **δέντρο ορφανών πλανητών** του γαλαξία όπου ανήκει το προς διαγραφή ηλιακό σύστημα.

➤ **O `<oid>` `<pid>`:** Γεγονός τύπου *Planet-Orphan Crash* το οποίο σηματοδοτεί τη σύγκρουση του ορφανού πλανήτη με αναγνωριστικό `<oid>` με τον πλανήτη με αναγνωριστικό `<pid>`. Η σύγκρουση έχει ως αποτέλεσμα την καταστροφή και των δύο πλανητών. Συγκεκριμένα, ο ορφανός πλανήτης με αναγνωριστικό `<oid>` διαγράφεται από το **δέντρο ορφανών πλανητών** στο οποίο ανήκει, αλλά και ο πλανήτης με αναγνωριστικό `<pid>` διαγράφεται από τη **δέντρο πλανητών** (planets) του ηλιακού συστήματος όπου ανήκει. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
O <oid> <pid>
  Planets =
    <pid1> : <distance1>
    <pid2> : <distance2>
    ...
    <pidn> : <distancen>
  Orphans = <oid1>, <oid2>, ... <oidm>
DONE
```

όπου n είναι το πλήθος των πλανητών που υπάρχουν στο ηλιακό σύστημα όπου ανήκει ο πλανήτης με αναγνωριστικό `<pid>`, m είναι το πλήθος ορφανών πλανητών στο **δέντρο ορφανών πλανητών** του γαλαξία στον οποίο ανήκει ο ορφανός πλανήτης με αναγνωριστικό `<oid>` και:

- για κάθε i , $1 \leq i \leq n$:
 - $\langle \text{pid}_i \rangle$ είναι το αναγνωριστικό του i -οστού πλανήτη στο ηλιακό σύστημα που περιείχε τον πλανήτη με αναγνωριστικό $\langle \text{pid} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου αυτού),
 - $\langle \text{distance}_i \rangle$ είναι η απόσταση του πλανήτη με αναγνωριστικό $\langle \text{pid}_i \rangle$ από το αστέρι-ήλιο του ηλιακού συστήματος που περιείχε τον πλανήτη με αναγνωριστικό $\langle \text{pid} \rangle$,
 - για κάθε i , $1 \leq i \leq m$, $\langle \text{oid}_i \rangle$ είναι το αναγνωριστικό του i -οστού ορφανού πλανήτη στο **δέντρο ορφανών πλανητών** του γαλαξία όπου ανήκε ο ορφανός πλανήτης με αναγνωριστικό $\langle \text{oid} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου αυτού).
- **C** $\langle \text{sid1} \rangle \langle \text{distance} \rangle \langle \text{sid2} \rangle \langle \text{sid3} \rangle$: Γεγονός τύπου *Binary Star Creation* το οποίο σηματοδοτεί τη δημιουργία δύο πλανητικών συστημάτων από το διαχωρισμό ενός ηλιακού συστήματος. Η δημιουργία δίδυμου συστήματος πλανητών έχει ως αποτέλεσμα το ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid1} \rangle$, το οποίο προϋπήρχε στον γαλαξία του, να διαχωριστεί σε δύο νέα ηλιακά συστήματα με αναγνωριστικά $\langle \text{sid2} \rangle$ και $\langle \text{sid3} \rangle$, αντίστοιχα. Μετά το διαχωρισμό του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid1} \rangle$, οι πλανήτες που απείχαν λιγότερο από $\langle \text{distance} \rangle$ km από το αστέρι-ήλιο θα ανήκουν στο ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid2} \rangle$, ενώ οι πλανήτες σε μαγαλύτερη απόσταση θα ανήκουν στο ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid3} \rangle$. Ο διαχωρισμός του δένδρου πλανητών του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid1} \rangle$ σε δύο δένδρα πλανητών τα οποία θα αντιστοιχούν στα δύο νέα ηλιακά συστήματα που θα δημιουργηθούν θα πρέπει να πραγματοποιείται με τον αλγόριθμο που ζητείται στην άσκηση 2 β) της 3^{ης} σειράς ασκήσεων. Επομένως, η χρονική πολυπλοκότητα του διαχωρισμού αυτού θα πρέπει να είναι $O(h)$, όπου h είναι το ύψος του δένδρου πλανητών του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid1} \rangle$. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:


```

C <sid1> <distance> <sid2> <sid3>
  sid1 Planets =
    <pid11> : <distance11>
    <pid12> : <distance12>
    ...
    <pid1k> : <distance1k>
  sid2 Planets =
    <pid21> : <distance21>
    <pid22> : <distance22>
    ...
    <pid2n> : <distance2n>
  sid3 Planets =
    <pid31> : <distance31>
    <pid32> : <distance32>
    ...
    <pid3m> : <distance3m>
DONE

```

όπου k είναι το πλήθος των πλανητών του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid1} \rangle$, n είναι το πλήθος των πλανητών του νέου ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid2} \rangle$, m είναι το πλήθος των πλανητών του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid3} \rangle$ και:

- για κάθε i , $1 \leq i \leq k$:
 - $\langle \text{pid}_{1i} \rangle$ είναι το αναγνωριστικό του i -οστού πλανήτη στο ηλιακό σύστημα με αναγνωριστικό $\langle \text{sid1} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου πλανητών του ηλιακού συστήματος),
 - $\langle \text{distance}_{1i} \rangle$ είναι η απόσταση του πλανήτη με αναγνωριστικό $\langle \text{pid}_{1i} \rangle$ από το αστέρι-ήλιο του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid1} \rangle$,
- για κάθε i , $1 \leq i \leq n$:
 - $\langle \text{pid}_{2i} \rangle$ είναι το αναγνωριστικό του i -οστού πλανήτη στο ηλιακού σύστημα με αναγνωριστικό $\langle \text{sid2} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου πλανητών του ηλιακού συστήματος),
 - $\langle \text{distance}_{2i} \rangle$ είναι η απόσταση του πλανήτη με αναγνωριστικό $\langle \text{pid}_{2i} \rangle$ από το αστέρι-ήλιο του ηλιακού συστήματος με αναγνωριστικό $\langle \text{sid2} \rangle$,
- για κάθε i , $1 \leq i \leq m$:
 - $\langle \text{pid}_{3i} \rangle$ είναι το αναγνωριστικό του i -οστού πλανήτη στο ηλιακού σύστημα με αναγνωριστικό $\langle \text{sid3} \rangle$ (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου πλανητών του ηλιακού συστήματος),
 - $\langle \text{distance}_{3i} \rangle$ είναι η απόσταση του πλανήτη με αναγνωριστικό $\langle \text{pid}_{3i} \rangle$

από το αστέρι-ήλιο του ηλιακού συστήματος με αναγνωριστικό <sid3>.

- **L <pid>**: Γεγονός τύπου *Lookup Planet* το οποίο σηματοδοτεί την αναζήτηση του πλανήτη με αναγνωριστικό <pid> στο σύμπαν. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L <pid>
  <gid> : <sid> : <distance>, <pid_p> <pid_n>
DONE
```

όπου:

- <sid> είναι το αναγνωριστικό του ηλιακού συστήματος στο οποίο ανήκει ο πλανήτης με αναγνωριστικό <pid>,
- <gid> είναι το αναγνωριστικό του γαλαξία στον οποίο βρίσκεται το ηλιακό σύστημα με αναγνωριστικό <sid>,
- <distance> είναι η απόσταση του πλανήτη με αναγνωριστικό <pid> από το αστέρι/ήλιο του ηλιακού του συστήματος,
- <pid_p> είναι το αναγνωριστικό του πλανήτη που είναι ο αμέσως προηγούμενος πλανήτης του πλανήτη με αναγνωριστικό <pid> βάσει της σειράς εγγύτητας στο αστέρι-ήλιος
- <pid_n> είναι το αναγνωριστικό του πλανήτη που είναι ο αμέσως επόμενος πλανήτης του πλανήτη με αναγνωριστικό <pid> βάσει της σειράς εγγύτητας στο αστέρι-ήλιος.

Η εύρεση των <pid_p> και <pid_n> θα πρέπει να επιτυγχάνεται σε χρονική πολυπλοκότητα $O(h)$ όπου h είναι το ύψος του δένδρου πλανητών που περιέχει τον πλανήτη με αναγνωριστικό <pid>.

- **K <oid>**: Γεγονός τύπου *Lookup Orphan Planet* το οποίο σηματοδοτεί την αναζήτηση του ορφανού πλανήτη με αναγνωριστικό <oid> στο σύμπαν. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
K <oid>
  <gid> <oid_p> <oid_n>
DONE
```

όπου <gid> είναι το αναγνωριστικό του γαλαξία στον οποίο βρίσκεται ο ορφανός πλανήτης με αναγνωριστικό <oid> και <oid_p> και <oid_n> είναι ο προηγούμενος και ο επόμενος πλανήτης στη διάταξη που προκύπτει αν εφαρμοστεί ενδοδιατεταγμένη διάσχιση στο δένδρο ορφανών πλανητών του γαλαξία με αναγνωριστικό <gid>. Η εύρεση των <oid_p> και <oid_n> θα πρέπει να πραγματοποιείται κατά την εύρεση του <oid> και επομένως **το επιπρόσθετο κόστος** στην χρονική πολυπλοκότητα του γεγονότος αυτού για την εύρεση των <oid_p> και <oid_n> θα πρέπει να είναι $O(1)$.

- **H <sid>**: Γεγονός τύπου *Print Solar* το οποίο σηματοδοτεί την εκτύπωση όλων των πλανητών στο ηλιακό σύστημα με αναγνωριστικό <sid>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
H <sid>
  Planets =
    <pid1> : <distance1>
    <pid2> : <distance2>
    ...
    <pidn> : <distancen>
DONE
```

όπου n είναι το πλήθος των πλανητών στο ηλιακό σύστημα με αναγνωριστικό <sid> και για κάθε i , $1 \leq i \leq n$:

- <pid _{i} > είναι το αναγνωριστικό του i -οστού πλανήτη (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου πλανητών του ηλιακού συστήματος με αναγνωριστικό <sid>) και
- <distance _{i} > είναι η απόσταση του πλανήτη με αναγνωριστικό <pid _{i} > από το αστέρι/ήλιο του ηλιακού συστήματος,

- **I <gid>**: Γεγονός τύπου *Print Orphans* το οποίο σηματοδοτεί την εκτύπωση όλων των ορφανών πλανητών του γαλαξία με αναγνωριστικό <gid>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
I <gid>
  Orphans =
    <oid1>
    <oid2>
    ...
    <oidn>
DONE
```

όπου n είναι το πλήθος των ορφανών πλανητών του γαλαξία με αναγνωριστικό <gid> και για κάθε i , $1 \leq i \leq n$, <oid _{i} > είναι το αναγνωριστικό του i -οστού ορφανού πλανήτη (στη διάταξη που προκύπτει από την ενδοδιατεταγμένη διάσχιση του δένδρου ορφανών πλανητών του γαλαξία με αναγνωριστικό <gid>).

- **J <gid>**: Γεγονός τύπου *Print Galaxy* το οποίο σηματοδοτεί την εκτύπωση όλων των ηλιακών συστημάτων και ορφανών πλανητών στον γαλαξία με αναγνωριστικό <gid>. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```

J <gid>
  Solars =
  H <sid1>
  H <sid2>
  ...
  H <sidn>

  Orphan planets =
  I <gid>
DONE

```

όπου n είναι το πλήθος των ηλιακών συστημάτων στο γαλαξία με αναγνωριστικό $\langle \text{gid} \rangle$, m είναι το πλήθος των ορφανών πλανητών στο γαλαξία με αναγνωριστικό $\langle \text{gid} \rangle$, για κάθε j , $1 \leq j \leq n$:

- sid_j είναι το αναγνωριστικό του j -οστού ηλιακού συστήματος στο γαλαξία με αναγνωριστικό $\langle \text{gid} \rangle$,
- όπου $H \text{ } \langle \text{sid}_j \rangle$ τυπώνεται η πληροφορία που θα τυπωνόταν αν εκτελούνταν το γεγονός *Print Solar* με παράμετρο $\langle \text{sid}_j \rangle$,
- όπου $I \text{ } \langle \text{gid} \rangle$ τυπώνεται η πληροφορία που θα τυπωνόταν αν εκτελούνταν το γεγονός *Print Orphans* με παράμετρο $\langle \text{gid} \rangle$.

- **U:** Γεγονός τύπου *Print Universe* το οποίο σηματοδοτεί την εκτύπωση όλων των γαλαξιών στο σύμπαν. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```

U
  J <gid1>
  J <gid2>
  ...
  J <gidn>
DONE

```

όπου n είναι το πλήθος των γαλαξιών στο σύμπαν και για κάθε i , $1 \leq i \leq n$:

- $\langle \text{gid}_i \rangle$ είναι το αναγνωριστικό του i -οστού γαλαξία στο σύμπαν,
- όπου $J \text{ } \langle \text{gid}_i \rangle$ τυπώνεται η πληροφορία που θα τυπωνόταν αν εκτελούνταν το γεγονός *Print Galaxy* με παράμετρο $\langle \text{gid}_i \rangle$.

- **E:** Γεγονός τύπου *End of World* το οποίο σηματοδοτεί την συντέλεια του κόσμου. Η συντέλεια του κόσμου έχει ως αποτέλεσμα τη διαγραφή όλων των δομών δεδομένων που χρησιμοποιούνται από το πρόγραμμα. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

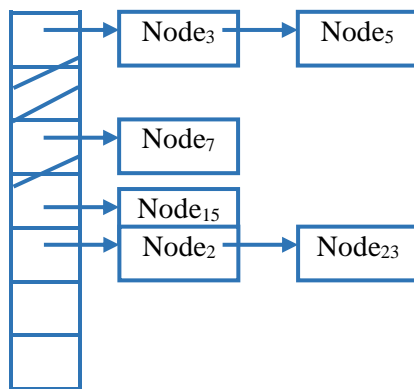
```

E DONE

```

ΓΕΓΟΝΟΤΑ BONUS

- **[Bonus 15%] Bonus 1:** Υλοποίηση της δομής των ορφανών πλανητών ενός γαλαξία με **πίνακα κατακερματισμού**. Το πεδίο `orphans` της εγγραφής `galaxie_t` θα πρέπει να είναι ένας πίνακας ο οποίος θα αποθηκεύει πληροφορίες για τους ορφανούς πλανήτες ενός γαλαξία. Η δομή αυτή ονομάζεται *πίνακας κατακερματισμού ορφανών πλανητών του εκάστοτε γαλαξία*. Το `orphans[k]` περιέχει έναν δείκτη (τύπου `planet_t*`) προς το πρώτο στοιχείο μιας αλυσίδας, **ταξινομημένης ως προς το αναγνωριστικό του κάθε ορφανού πλανήτη**, στην οποία αποθηκεύονται ορφανοί πλανήτες τέτοιοι ώστε η τιμή κατακερματισμού του αναγνωριστικού του πλανήτη είναι ίση με k . Το μέγεθος του πίνακα κατακερματισμού (`ahashable_size_g`) θα πρέπει να επιλέγεται από εσάς προσεχτικά και θα πρέπει να είστε σε θέση να δικαιολογήσετε την επιλογή σας. Για την υλοποίηση της συνάρτησης κατακερματισμού θα πρέπει να χρησιμοποιήσετε καθολικό κατακερματισμό. Για την υλοποίηση του καθολικού κατακερματισμού παρέχονται, ένας πίνακας `primes_g[]` με 160 πρώτους αριθμούς σε αύξουσα σειρά και το μέγιστο κλειδί που μπορεί να έχει οποιοσδήποτε ορφανός πλανήτης, μέσω της μεταβλητής `max_id_g`. Οι μεταβλητές αυτές είναι `global` και έχουν δηλωθεί στο `myuniverse.h`. Αρχικοποιούνται στη `main` βάσει τιμών που αναγράφονται στις πρώτες γραμμές του `test_file`.

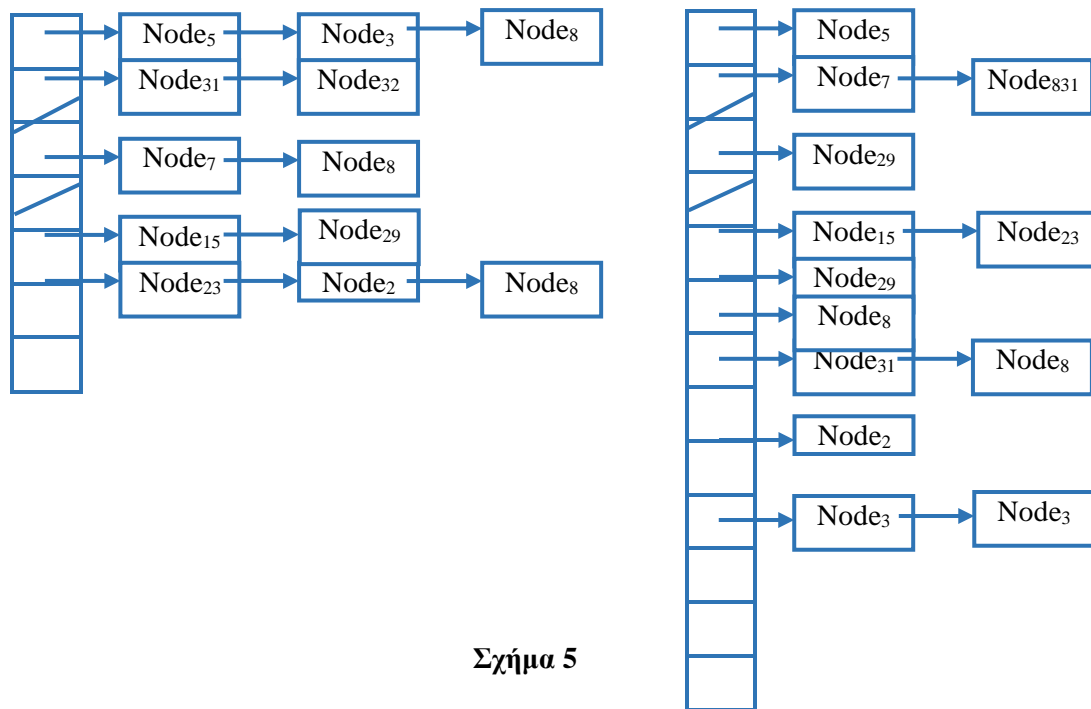


Σχήμα 4

[Bonus 10%] Bonus 2: Για κάθε πίνακα κατακερματισμού (που περιγράφεται στο Bonus 1), υλοποιήστε επιπρόσθετα τη **λειτουργία επέκτασης**. Το μέγεθος του πίνακα κατακερματισμού θα πρέπει να επαυξάνεται ή να μειώνεται κατά τη διάρκεια εκτέλεσης του προγράμματος ανάλογα με το πλήθος των κόμβων που υπάρχουν σε αυτόν. Η αύξηση του μεγέθους του πίνακα θα γίνεται ως εξής. Στο `myuniverse.h` θα υπάρχει ένας πίνακας `primes_g[]` από πρώτους αριθμούς, τα στοιχεία του οποίου θα έχουν αρχικοποιηθεί από τη `main` (όπως περιγράφεται στο **Bonus 1**). Την i -οστή φορά που πρέπει να επαυξηθεί το μέγεθος του πίνακα κατακερματισμού, θα πρέπει, μετά την επαύξηση, αυτός να έχει μέγεθος ίσο με τον πρώτο αριθμό που είναι αποθηκευμένος στο στοιχείο `primes_g[i]`. Στο παρακάτω σχήμα (Σχήμα 5) φαίνεται ο τρόπος με τον οποίο θα πραγματοποιείται η διαδικασία αυτή.

Όταν το πλήθος των κόμβων κάποιας λίστας γίνει ίσο με ένα πάνω φράγμα `HT_UPPER`, τότε το μέγεθος του πίνακα κατακερματισμού πρέπει να αυξηθεί (με τη διαδικασία που περιγράφηκε παραπάνω) και να υπολογιστεί ξανά η θέση του κάθε κλειδιού χρησιμοποιώντας τη συνάρτηση κατακερματισμού. Αν το πλήθος των κόμβων **σε κάθε λίστα** γίνει μικρότερο από κάποιο κάτω φράγμα, `HT_LOWER`, θα πρέπει το μέγεθος του πίνακα κατακερματισμού να μειώνεται ακολουθώντας την αντίστροφη διαδικασία. Συγκεκριμένα, αν το τρέχον μέγεθος του πίνακα είναι ίσο με `primes_g[i]`, μετά τη μείωση, το μέγεθος θα πρέπει να είναι ίσο με `primes_g[i-1]`. Οι μεταβλητές `HT_UPPER` και `HT_LOWER` είναι `global` και έχουν δηλωθεί

στο myuniverse.h. Αρχικοποιούνται στη main βάσει τιμών που αναγράφονται στην πρώτη γραμμή του test_file.



Σχήμα 5

ΠΡΟΣΟΧΗ

Κάθε υλοποίηση κάποιου bonus πρέπει να γίνει σε διαφορετικό αρχείο/α και ΟΧΙ στο βασικό/ά αρχείο/ά του project σας και θα πρέπει να κάνετε submit το βασικό project και όσα αρχεία αντιστοιχούν σε κάθε ένα από τα bonus που θα υλοποιήσετε.

Δομές Δεδομένων

Στη συνέχεια παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας.

```

////////////////////////////////////
//                               Declaring the needed structures
////////////////////////////////////
typedef struct galaxy   galaxy_t;
typedef struct solar     solar_t;
typedef struct planet    planet_t;
typedef struct ocluster ocluster_t;

////////////////////////////////////
//                               Defining the needed structures
////////////////////////////////////

/**
 * Structure defining a node of the planets list (lista planitwn)
 */
struct ocluster {
    int      cid;                /**< orphan cluster identifier > 0 */
    planet_t *orphans;          /**< Pointer to the first node in the
                                * orphan planets list
                                * (lista orfanwn planitwn) */
};

/**
 * Structure defining a node of the galaxies array (pinakas gala3iwn)
 */
struct galaxy {
    int      gid;                /**< The galaxy identifier. >0 */
    solar_t  *solars;           /**< Pointer to first element in
                                * solar systems list (lista iliakwn
                                * sistimatwn) */
    planet_t *orphans;          /**< The orphan tree
                                * (dentro orfanwn planitwn) */
    solar_t  *ssentinel;        /**< Pointer to the sentinel node
                                * (komvo frouro) of solar systems
                                * list (lista iliakwn sistimatwn) */
};

/**
 * Structure defining a node of the solar systems list (lista iliakwn sistimatwn)
 */
struct solar {
    int      sid;                /**< Solar system identifier. >0 */
    planet_t *planets;           /**< Pointer to planets tree
                                * (dentro planitwn) */
    plante_t *psentinel         /**< pointer to the sentinel node
                                * (komvo frouro) of planetet tree
                                * list (lista iliakwn sistimatwn) */
    solar_t  *next;              /**< Pointer to next node in
                                * solar systems list
                                * (lista iliakwn sistimatwn) */
};

/**
 * Structure defining a node of the planets list (lista planitwn)
 */
struct planet {
    int      pid;                /**< Planet identifier. >0 */
    int      distance;           /**< Distance from star in solar system */
    planet_t *lc;                /**< Pointer to left child */
    planet_t *rc;                /**< Pointer to right child */
    planet_t *parent;            /**< Pointer to parent */
};

```



```
};
```

```
// For simplicity we define the Galaxies array as a global variable
galaxy_t Galaxies[MAX_GALAXIES];          /**< The galaxies array (pinakas gala3iwn).
                                           * This is an array of lists */
```

Συμβουλές

Για λόγους ευκολότερης αποσφαλμάτωσης του κώδικα που θα δημιουργήσετε, συνίσταται ισχυρά κατά την διαγραφή κόμβων να αναθέτετε την τιμή NULL στους δείκτες του προς διαγραφή στοιχείου. Ακόμη συνίσταται να ανατίθεται την τιμή INT_MAX στα υπόλοιπα πεδία του κόμβου που είναι τύπου (int).