ΗΥ240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2011-12
Διδάσκουσα: Παναγιώτα Φατούρου

Προγραμματιστική Εργασία
2ο Μέρος

Ημερομηνία Παράδοσης: Παρασκευή, 14 Δεκεμβρίου 2012, ώρα 23:59.
Τρόπος Παράδοσης: Χρησιμοποιώντας το πρόγραμμα submit. Πληροφορίες για το πώς λειτουργεί το submit παρέχονται στην ιστοσελίδα του μαθήματος.

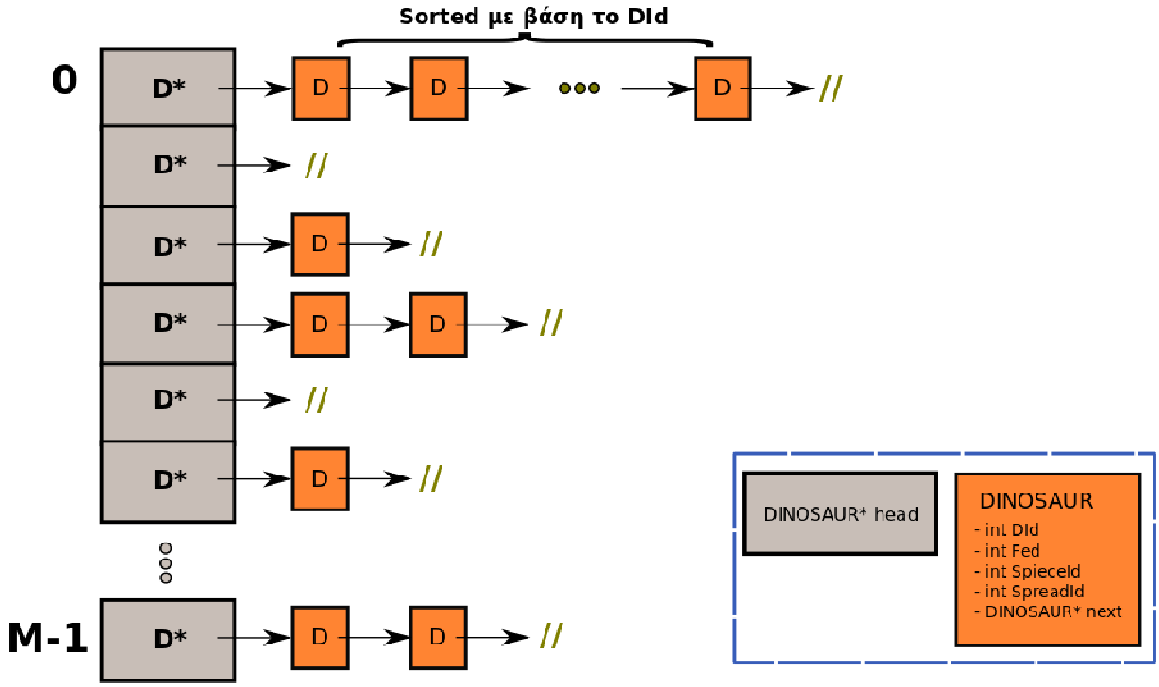
Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Για κάθε δεινόσαυρο υπάρχει ένα struct Dinosaur με τα εξής πεδία (προσέξτε πως έχει προστεθεί ένα ακόμη πεδίο σε αυτή τη δομή, συγκριτικά με το πρώτο μέρος της προγραμματιστικής εργασίας):

- DId: αναγνωριστικό του δεινόσαυρου, μοναδικό για κάθε δεινόσαυρο, ανεξαρτήτως αν αυτός είναι χορτοφάγος ή σαρκοφάγος και ανεξαρτήτως του είδους στο οποίο ανήκει ο δεινόσαυρος.
- Fed: το πεδίο αυτό είναι 0 αν ο δεινόσαυρος είναι χορτοφάγος και 1 αν είναι σαρκοφάγος.
- SpieceId: ένας ακέραιος (*αναγνωριστικό είδους*) που δηλώνει το είδος του δεινόσαυρου.
- SpreadId: αναγνωριστικό του κοπαδιού στο οποίο ανήκει ο δεινόσαυρος.
- Age: ηλικία του δεινόσαυρου.
- Strength: αριθμός που εκφράζει τη δύναμη του δεινόσαυρου. Ο αριθμός αυτός μειώνεται κάθε φορά που ο δεινόσαυρος εμπλέκεται σε έναν πόλεμο, ενώ δεινόσαυροι που δεν έχουν αρκετά μεγάλη δύναμη για να ανταπεξέλθουν στον πόλεμο πεθαίνουν.
- next: δείκτης στο επόμενο στοιχείο της αλυσίδας του πίνακα κατακερματισμού στην οποία ανήκει ο δεινόσαυρος.

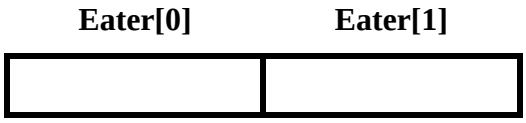
Το πρόγραμμα σας θα πρέπει να διατηρεί έναν **πίνακα κατακερματισμού** DIN[M], ο οποίος περιέχει πληροφορίες για όλους τους δεινόσαυρους στη Παγγαία. Κάθε στοιχείο DIN[i] του πίνακα αυτού περιέχει έναν δείκτη στο πρώτο στοιχείο μιας αλυσίδας που περιέχει εκείνα τα struct Dinosaur για τα οποία το id τους έχει τιμή κατακερματισμού ίση με i. Άρα, ο πίνακας κατακερματισμού επιλύει συγκρούσεις με τη μέθοδο των αλυσίδων, όπου οι αλυσίδες είναι ταξινομημένες ως προς το αναγνωριστικό (Did) των δεινοσαύρων. Το μέγεθος του πίνακα κατακερματισμού M θα πρέπει να επιλέγεται από εσάς προσεχτικά και θα πρέπει να είστε σε θέση να δικαιολογήσετε την επιλογή σας. Το ίδιο θα πρέπει να ισχύει και για τη συνάρτηση κατακερματισμού που θα χρησιμοποιήσετε.

Ο πίνακας κατακερματισμού δεινοσαύρων παρουσιάζεται στο Σχήμα 1.



Σχήμα 1

Όπως προαναφέρθηκε, οι δεινόσαυροι κατηγοριοποιούνται είτε ως φυτοφάγοι ή ως σαρκοφάγοι. Πληροφορίες για κάθε κατηγορία αποθηκεύονται στα στοιχεία ενός πίνακα Eater[2], όπως φαίνεται στο Σχήμα 2. Ο πίνακας αυτός ονομάζεται *πίνακας διατροφικών κατηγοριών δεινοσαύρων*.



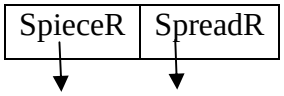
Σχήμα 2

Τα στοιχεία του πίνακα Eater[2] είναι τύπου Feed (struct Feed). Κάθε ένα από αυτά αποτελείται από τα εξής πεδία:

- Έναν δείκτη, SpieceR, ο οποίος δείχνει στη ρίζα ενός **διπλά-συνδεδεμένου, ταξινομημένου,** ως προς το SpieceId, **δένδρου** (δηλαδή **διπλά-συνδεδεμένου δένδρου δυαδικής αναζήτησης**) **με κόμβο φρουρό** που ονομάζεται *δένδρο ειδών δεινοσαύρων*. Κάθε κόμβος του δένδρου περιέχει πληροφορίες για κάποιο είδος δεινόσαυρων.
- Έναν δείκτη, SpreadR, ο οποίος δείχνει στη ρίζα ενός **διπλά-συνδεδεμένου, προδιατεταγμένα-ταξινομημένου** (δες άσκηση 3 του 3^{ου} σετ ασκήσεων)**,** ως προς το SpreadId, **δένδρου** που ονομάζεται *δένδρο κοπαδιών δεινοσαύρων*. Κάθε κόμβος του

δένδρου αυτού περιέχει πληροφορίες για κάποιο κοπάδι δεινοσαύρων. Το δένδρο αυτό δεν έχει κόμβο φρουρό.

Η μορφή κάθε στοιχείου του πίνακα διατροφικών κατηγοριών παρουσιάζεται στο Σχήμα 3.



Σχήμα 3

Κάθε κόμβος του δένδρου ειδών δεινοσαύρων είναι ένα struct τύπου SpieceNode που αποτελείται από τα ακόλουθα πεδία:

- Έναν ακέραιο SpieceId που αποτελεί το αναγνωριστικό του είδους.
- Έναν δείκτη SpieceDp που δείχνει στον πρώτο στοιχείο μιας λίστας δεινοσαύρων που ανήκουν σε αυτό το είδος. Η λίστα αυτή λέγεται *λίστα δεινοσαύρων του τύπου SpieceId*.
- Έναν δείκτη p που δείχνει στον πατέρα του κόμβου.
- Έναν δείκτη lc που δείχνει στο αριστερό παιδί του κόμβου.
- Έναν δείκτη rc που δείχνει στο δεξιό παιδί του κόμβου.

Κάθε στοιχείο της λίστας δεινοσαύρων τύπου SpieceId περιέχει στοιχεία τύπου struct SpieceDin. Το struct αυτό έχει τα ακόλουθα πεδία:

- int DId: το αναγνωριστικό του δεινόσαυρου (προσέξτε πως αυτό το πεδίο έχει αλλάξει συγκριτικά με το 1^{ο} μέρος της προγραμματιστικής εργασίας).
- struct SpieceDin *next: δείκτης στο επόμενο στοιχείο της λίστας.

Η λίστα δεινοσαύρων τύπου SpieceId είναι **μη ταξινομημένη** και υλοποιεί το ευριστικό Μετακίνησης στην Αρχή (move to front).

Ομοίως, κάθε κόμβος του δένδρου κοπαδιών δεινοσαύρων είναι ένα struct που αποτελείται από τα ακόλουθα πεδία:

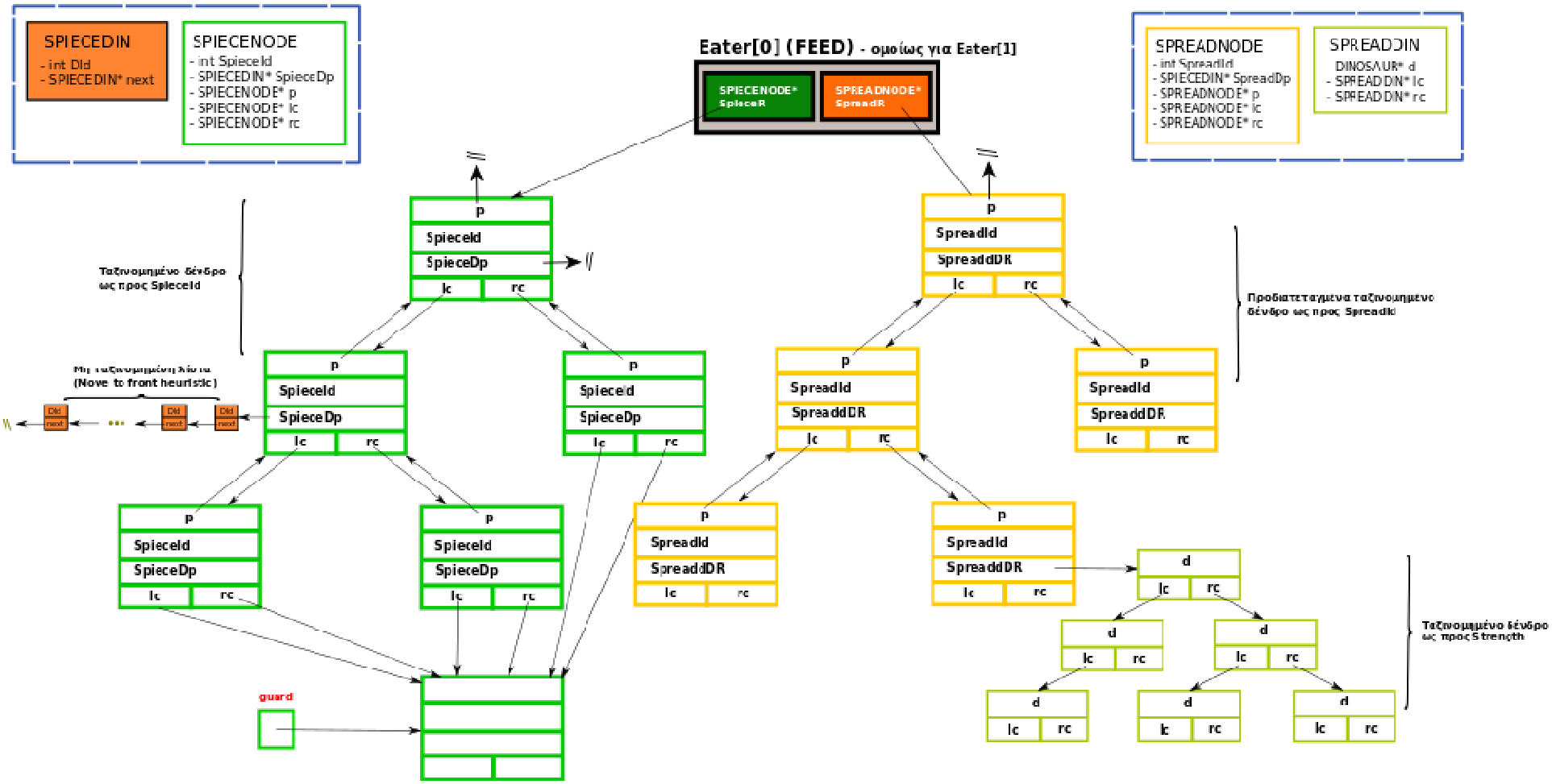
- Έναν ακέραιο spreadId που αποτελεί το αναγνωριστικό του κοπαδιού.
- Έναν δείκτη spreadDR που δείχνει στη ρίζα ενός ταξινομημένου δένδρου δεινοσαύρων που ανήκουν σε αυτό το κοπάδι. Το δένδρο αυτό, που ονομάζεται δένδρο *του κοπαδιού SpreadId*, είναι (ενδοδιατεταγμένα) ταξινομημένο ως προς τη δύναμη (Strength) των δεινοσαύρων (είναι δηλαδή δένδρο δυαδικής αναζήτησης), δεν είναι διπλά-συνδεδεμένο και δεν έχει κόμβο φρουρό.
- Έναν δείκτη p που δείχνει στον πατέρα του κόμβου.
- Έναν δείκτη lc που δείχνει στο αριστερό παιδί του κόμβου.
- Έναν δείκτη rc που δείχνει στο δεξί παιδί του κόμβου.

Κάθε κόμβος του δένδρου του κοπαδιού SpreadId περιέχει στοιχεία τύπου struct SpreadDin. Το struct αυτό έχει τα ακόλουθα πεδία:

- struct Dinosaur *d: δείκτης σε struct τύπου Dinosaur που περιέχει πληροφορίες για τον δεινόσαυρο.

- struct SpreadDin *lc: δείκτης στο αριστερό παιδί του κόμβου.
- struct SpreadDin *rc: δείκτης στο δεξιό παιδί του κόμβου.

Οι δομών δεδομένων που περιγράφονται παραπάνω παρουσιάζονται στο Σχήμα 4.


SPIECEDIN
- int DId
- SPIECEDIN* next
SPIECENODE
- int SpieceId
- SPIECEDIN* SpieceDp
- SPIECENODE* p
- SPIECENODE* lc
- SPIECENODE* rc
Eater[0] (FEED) - ομοίως για Eater[1]
SPIECENODE*
SpieceR
SPREADNODE*
SpreadR
SPREADNODE
- int SpreadId
- SPIECEDIN* SpreadDp
- SPREADNODE* p
- SPREADNODE* lc
- SPREADNODE* rc
SPREADDIN
DINOSAUR* d
- SPREADDIN* lc
- SPREADDIN* rc
Ταξινομημένο δένδρο ως προς SpieceId
Προδιατεταγμένα ταξινομημένο δένδρο ως προς SpreadId
Ταξινομμένο δένδρο ως προς Strength
(Move to front heuristic)
guard
p
SpieceId
SpieceDp
SpreadId
SpreaddDR
d
lc
rc

Τρόπος Λειτουργίας Προγράμματος

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

<name of executable file> <input-file> <MAX_AGE> <M>

όπου <name of executable file> είναι το όνομα του εκτελέσιμου αρχείου του προγράμματος (π.χ., a.out), <MAX_AGE> είναι ένας ακέραιος που υποδηλώνει πόσα το πολύ χρόνια ζει κάθε δεινόσαυρος, <M > είναι είναι το μέγεθος του πίνακα κατακερματισμού και <input-file> είναι το όνομα ενός αρχείου εισόδου που περιέχει γεγονότα των ακόλουθων μορφών:

- C <N> <Fed> <SpieceId> <SpreadId> <Id_1> <Id_2> ... <Id_N> <$Strength_1$> <$Strength_2$> ... <$Strength_N$>: Γεγονός τύπου Create το οποίο σηματοδοτεί τη δημιουργία <N> νέων δεινοσαύρων στο οικοσύστημα με αναγνωριστικά <Id_1>, …, <Id_N> και δύναμη <$Strength_1$> <$Strength_2$> ... <$Strength_N$>, αντίστοιχα. Το πεδίο <Fed> είναι 0 ή 1 και παρέχει την πληροφορία του αν οι νέοι δεινόσαυροι είναι χορτοφάγοι ή σαρκοφάγοι, αντίστοιχα. Το πεδίο <SpieceId> είναι το αναγνωριστικό του είδους στο οποίο ανήκουν οι νέοι δεινόσαυροι. Επιπρόσθετα, οι νέοι δεινόσαυροι ανήκουν στο κοπάδι με αναγνωριστικό <SpreadId>. Επομένως, τα <N>, <Fed>, <SpieceId>, και <SpreadId> είναι ακέραιοι αριθμοί που παρέχουν τις παραπάνω πληροφορίες για το δεινόσαυρο. Αν το <SpieceId> ισούται με -1, τότε δημιουργούνται δεινόσαυροι όλων των ειδών με τον ακόλουθο τρόπο. Αν υπάρχουν k είδη δεινοσαύρων στο οικοσύστημα με τις δεδομένες διατροφικές συνήθειες, δημιουργούνται N/k νέοι δεινόσαυροι σε κάθε είδος (οι πρώτοι $\lfloor N/k \rfloor$ από αυτούς έχουν αναγνωριστικά <Id_1>, <Id_2>, …, <$Id_{\lfloor N/k \rfloor}$>, οι επόμενοι $\lfloor N/k \rfloor$ έχουν αναγνωριστικά <$Id_{\lfloor N/k \rfloor+1}$>, <$Id_{\lfloor N/k \rfloor+2}$>, …, <$Id_{2\lfloor N/k \rfloor}$>, κ.ο.κ) Όλοι ανήκουν ωστόσο στο ίδιο κοπάδι και όλοι έχουν τις ίδιες διατροφικές συνήθειες. Η ηλικία των δεινοσαύρων πρέπει να επιλέγεται με τυχαίο τρόπο (χρησιμοποιώντας την rand()), λαμβάνοντας υπ' όψιν ότι η μέγιστη ηλικία είναι MAX_AGE.

  Ο πίνακας κατακερματισμού δεινοσαύρων θα πρέπει να ενημερωθεί προκειμένου να περιέχει τους νέους δεινόσαυρους.

  Επιπρόσθετα, αν <SpieceId> ≠ -1 και ο δεινόσαυρος ανήκει σε κάποιο είδος που δεν υπάρχει στο σύστημα, θα πρέπει να προστεθεί μια νέα εγγραφή στο δένδρο ειδών δεινοσαύρων και <N> νέες εγγραφές στη λίστα δεινοσαύρων του είδους στο οποίο ανήκει ο δεινόσαυρος. Αν <SpieceId> = -1, για κάθε είδος του δένδρου ειδών δεινοσαύρων του Eater[<Fed>] πρέπει να γίνουν N/k εισαγωγές στη λίστα δεινοσαύρων του είδους.

  Αλλαγές θα πρέπει να γίνουν στο δένδρο κοπαδιών δεινοσαύρων καθώς και στο δένδρο του κοπαδιού στο οποίο ανήκουν οι νέοι δεινόσαυροι. Οι αλλαγές αυτές πρέπει να γίνουν με κατάλληλο τρόπο ώστε τα δένδρα αυτά να μην χάνουν τις ιδιότητες που έχουν (π.χ., την ταξινόμηση).

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

  C <N> <Fed> <SpieceId> <SpreadId> <Id_1> <Id_2> ... <Id_N> <$Strength_1$> <$Strength_2$> ... <$Strength_N$> DONE

  Eater[0]

  SpieceTREE = <SpieceId1>,

  List of Spiece <SpieceId1> = <d_1^1, d_2^1, …, d_{r1}^1>

  < SpieceId2>,

  List of Spiece <SpieceId2> = <d_1^2, d_2^2, …, d_{r2}^2>

  …

  …

  <SpieceIdk>,

List of Spiece <SpieceIdk> = <d_1^k, d_2^k, …, d_{rk}^k>

SpreadTree = <SpreadId1>,

Dinosaurs of Spread <SpreadId1> = <$d'_1{}^1$, $d'_2{}^1$, …, $d'_{m1}{}^1$>

< SpreadId2>,

Dinosaurs of Spread <SpreadId2> = <$d'_1{}^2$, $d'_2{}^2$, …, $d'_{m2}{}^2$>

…

…

<SpreadIdl>,

Dinosaurs of Spread <SpreadIdl> = <$d'_1{}^l$, $d'_2{}^l$, …, $d'_{ml}{}^l$>

Eater[1]

<τυπώνεται παρόμοια πληροφορία όπως και για την κατηγορία Eater[0]>

<η τύπωση της πληροφορίας πρέπει να ακολουθεί παρόμοιο format

όπως εκείνο της πληροφορίας για την κατηγορία Eater[0]>

SET OF DINOSAURS:

<DId_1, Fed_1, $SpieceId_1$, $SpreadId_1$, Age_1, $Strength_1$>

<DId_2, Fed_2, $SpieceId_2$, $SpreadId_2$, Age_2, $Strength_2$>

…

<DId_h, Fed_h, $SpieceId_h$, $SpreadId_h$, Age_h, $Strength_h$>

όπου k, l είναι το πλήθος στοιχείων στα δένδρα ειδών δεινοσαύρων και κοπαδιών, αντίστοιχα, της κατηγορίας Eater[0] και <d_1^i, d_2^i, …, d_{ri}^i>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στη λίστα δεινοσαύρων του είδους <$SpieceId_i$> (όπου έχουμε υποθέσει πως αυτά είναι r_i στο πλήθος). Επίσης, <$d'_1{}^i$, $d'_2{}^i$, …, $d'_{mi}{}^i$>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στο δένδρο δεινοσαύρων του κοπαδιού <$SpreadId_i$> (όπου έχουμε υποθέσει πως αυτά είναι m_i στο πλήθος).

Τέλος, h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \le i \le k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

- B <N> <DId> <Id_1> <Id_2> ... <Id_N> <$Strength_1$> <$Strength_2$> ... <$Strength_N$>: Γεγονός τύπου Born το οποίο σηματοδοτεί τη γέννηση <N> νέων δεινοσαύρων στο οικοσύστημα με αναγνωριστικά <Id_1>, <Id_2>, ..., <Id_N> και δύναμη <$Strength_1$>, <$Strength_2$>, ..., <$Strength_N$>, αντίστοιχα. Οι δεινόσαυροι αυτοί είναι παιδιά της δεινόσαυρου με αναγνωριστικό <DId> και άρα έχουν τις ίδιες διατροφικές συνήθειες, ανήκουν στο ίδιο κοπάδι και είναι του ιδίου είδους με τη μητέρα τους, δηλαδή με το δεινόσαυρο με αναγνωριστικό <DId>. Η ηλικία τους τη στιγμή της γέννησης είναι 0.

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

  B <N> <DId> <Id_1> <Id_2> ... <Id_N> <$Strength_1$> <$Strength_2$> ... <$Strength_N$> DONE

  Parent Dinosaur: Eater = <Fed>, Spc = <SpieceId>, Spr = <SpreadId>,Age = <Age>,Str = <Strength>

  SpieceLIST of <SpieceId> = <d_1, d_2, …, d_k>

  SpreadTree of <SpreadId> = <d'_1, d'_2, …, d'_r>

SET OF DINOSAURS:

<DId_1, Fed_1, $SpieceId_1$, $SpreadId_1$, Age_1, $Strength_1$>

<DId_2, Fed_2, $SpieceId_2$, $SpreadId_2$, Age_2, $Strength_2$>

…

<DId_h, Fed_h, $SpieceId_h$, $SpreadId_h$, Age_h, $Strength_h$>

όπου <Fed>, <SpieceId>, <SpreadId>, <Age>, <Strength> υποδηλώνουν τις διατροφικές συνήθειες, το είδος, το κοπάδι, την ηλικία και τη δύναμη του γονικού δεινόσαυρου. Επομένως, οι νέοι δεινόσαυροι ανήκουν στο είδος με αναγνωριστικό <SpieceId> και στο κοπάδι με αναγνωριστικό <SpreadId>. Επιπρόσθετα, k είναι το πλήθος στοιχείων στη λίστα δεινοσαύρων του είδους <SpieceId> και <d_1, d_2, …, d_k> είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στη λίστα αυτή. Επίσης, r είναι το πλήθος στοιχείων στο δένδρο δεινοσαύρων του κοπαδιού <SpreadId> και <d'_1, d'_2, …, d'_r> είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στο δένδρο αυτό.

Τέλος, h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \leq i \leq k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

- W <Fed1> <SpreadId1> <Fed2> <SpreadId2> <Strength>: Γεγονός τύπου War που σηματοδοτεί έναν διαπληκτισμό μεταξύ του κοπαδιού της διατροφικής κατηγορίας <Fed1> με αναγνωριστικό <SpreadId1> και εκείνου της διατροφικής κατηγορίας <Fed2> με αναγνωριστικό <SpreadId2>. Η δύναμη όλων των δεινόσαυρων και των δύο κοπαδιών μειώνεται κατά <Strength>. Οι δεινόσαυροι των οποίων η δύναμη μειώνεται σε μια τιμή μικρότερη ή ίση του μηδενός πεθαίνουν. Αν τα δύο κοπάδια αποτελούνται από δεινόσαυρους με τις ίδιες διατροφικές συνήθειες (π.χ. οι δεινόσαυροι και στα δύο κοπάδια είναι χορτοφάγοι), τα κοπάδια συνενώνονται. Διαφορετικά, τα κοπάδια παραμένουν με όσα μέλη τους έχουν απομείνει.

  Το κοπάδι με αναγνωριστικό <Spread1> θα πρέπει να αναζητηθεί στην διατροφική κατηγορία δεινοσαύρων Eater[<Fed1>]. Το δένδρο δεινοσαύρων του κοπαδιού <Spread1> διατρέχεται και για κάθε δεινόσαυρο η δύναμή του μειώνεται κατά <Strength>. Αν η δύναμη του δεινόσαυρου γίνει ίση ή μικρότερη του 0, ο δεινόσαυρος διαγράφεται. Αυτό πραγματοποιείται ως εξής: (1) ο δεινόσαυρος διαγράφεται από το δένδρο δεινοσαύρων του συγκεκριμένου κοπαδιού, (2) ο δεινόσαυρος διαγράφεται από τη λίστα δεινοσαύρων του είδους στο οποίο αυτός ανήκει, και τέλος (3) ο δεινόσαυρος διαγράφεται από τον πίνακα κατακερματισμού δεινοσαύρων. Παρόμοιες ενέργειες γίνονται για τους δεινόσαυρους του κοπαδιού με αναγνωριστικό <$Spread_2$> της διατροφικής κατηγορίας Eater[<Fed2>]. Αν το δένδρο του κοπαδιού <SpreadId1> ή το δένδρο του κοπαδιού <SpreadId2> είναι κενό μετά τις διαγραφές, το αντίστοιχο στοιχείο του δένδρου κοπαδιών δεινοσαύρων πρέπει να διαγραφεί. Διαγραφές στοιχείων ενδέχεται να χρειάζονται και στο δένδρο ειδών δεινοσαύρων, αν η λίστα για κάποιο συγκεκριμένο είδος μείνει κενή.

  Αν <Fed1> = <Fed2> τότε τα κοπάδια <SpreadId1> και <SpreadId2> πρέπει να συγχωνευτούν.

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

  W <Fed1> <SpreadId1> <Fed2> <SpreadId2> <Strength> DONE

  Eater[<Fed1>]

  SpieceTREE = <SpieceId1>,

List of Spiece <SpieceId1> = <d_1^1, d_2^1, …, d_{r1}^1>

< SpieceId2>,

List of Spiece <SpieceId2> = <d_1^2, d_2^2, …, d_{r2}^2>

…

…

<SpieceIdk>,

List of Spiece <SpieceIdk> = <d_1^k, d_2^k, …, d_{rk}^k>

SpreadTREE = <SpreadId1>,

Tree of Spread <SpreadId1> = <$d'_1{}^1$, $d'_2{}^1$, …, $d'_{m1}{}^1$>

< SpreadId2>,

Tree of Spread <SpreadId2> = <$d'_1{}^2$, $d'_2{}^2$, …, $d'_{m2}{}^2$>

…

…

<SpreadIdl>,

Tree of Spread <SpreadIdl> = <$d'_1{}^l$, $d'_2{}^l$, …, $d'_{ml}{}^l$>

Eater[<Fed2>]

<Αν <Fed1> ≠ <Fed2>, παρόμοια πληροφορία όπως και για την κατηγορία Eater[<Fed1>]>

<η τύπωση της πληροφορίας πρέπει να ακολουθεί παρόμοιο format

όπως εκείνο της πληροφορίας για την κατηγορία Eater[0]>

SET OF DINOSAURS:

<DId_1, Fed_1, $SpieceId_1$, $SpreadId_1$, Age_1, $Strength_1$>

<DId_2, Fed_2, $SpieceId_2$, $SpreadId_2$, Age_2, $Strength_2$>

…

<DId_h, Fed_h, $SpieceId_h$, $SpreadId_h$, Age_h, $Strength_h$>

όπου k, l είναι το πλήθος στοιχείων στα δένδρα ειδών δεινοσαύρων και κοπαδιών, αντίστοιχα, της διατροφικής κατηγορίας Eater[0] και <d_1^i, d_2^i, …, d_{ri}^i>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στη λίστα δεινοσαύρων του είδους <$SpieceId_i$> (όπου έχουμε υποθέσει πως αυτά είναι r_i στο πλήθος). Επίσης, <$d'_1{}^i$, $d'_2{}^i$, …, $d'_{mi}{}^i$>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στο δένδρο δεινοσαύρων του κοπαδιού <$SpreadId_i$> (όπου έχουμε υποθέσει πως αυτά είναι m_i στο πλήθος).

Τέλος, h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \le i \le k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

- S <Fed> <SpreadId> <Strength>: Γεγονός τύπου Split που σηματοδοτεί το διαχωρισμό του κοπαδιού με αναγνωριστικό <SpreadId> της διατροφικής κατηγορίας <Fed> σε δύο διαφορετικά κοπάδια. Συγκεκριμένα, οι δεινόσαυροι του κοπαδιού <SpreadId> που έχουν δύναμη μεγαλύτερη από <Strength> μένουν στο κοπάδι ενώ εκείνοι με μικρότερη δύναμη δημιουργούν ένα νέο κοπάδι.

(Προσέξτε πως τώρα ο διαχωρισμός γίνεται βάσει της δύναμης και όχι βάσει της ηλικίας όπως γινόταν στο 1° μέρος)

Οι ενέργειες που θα πρέπει να πραγματοποιηθούν είναι οι εξής: (1) εισαγωγή ενός νέου στοιχείου που θα αντιστοιχεί στο νέο κοπάδι στο δένδρο κοπαδιών δεινοσαύρων του Eater[<Fed>], (2) εισαγωγή των δεινοσαύρων που απαιτούνται στο δένδρο δεινοσαύρων του νέου κοπαδιού, και (3) αλλαγές θα πρέπει να γίνουν και στα struct Dinosaur των δεινοσαύρων αυτών.

Το γεγονός αυτό θα πρέπει να εκτελείται σε χρόνο O(h), όπου h είναι το ύψος του δένδρο του κοπαδιού με αναγνωριστικό <SpreadId> της διατροφικής κατηγορίας <Fed>. Για να επιτευχθεί αυτό, εκτελέστε μια αναζήτηση για το <Strength>, αποκόβοντας τους δείκτες που «κρέμονται» αριστερά και δεξιά του μονοπατιού. Στη συνέχεια, συγχωνεύστε τα δένδρα του προκύπτοντος δάσους.

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

S <Fed> <SpreadId> <Strength> DONE

Eater[<Fed>]

SpreadTREE = <SpreadId1>,

Tree of Spread <SpreadId1> = <d_1^1, d_2^1, …, d_{r1}^1>

< SpreadId2>,

Tree of Spread <SpreadId2> = <d_1^2, d_2^2, …, d_{r2}^2>

…

…

<SpreadIdk>,

Tree of Spread <SpreadIdk> = <d_1^k, d_2^k, …, d_{rk}^k>

SET OF DINOSAURS: <DId_1, $SpreadId_1$ >, <DId_2, $SpreadId_2$ >, …, <DId_h, $SpreadId_h$ >

όπου k είναι το πλήθος των στοιχείων στο δένδρο κοπαδιών δεινοσαύρων της διατροφικής κατηγορίας Eater[<Fed>] και <d_1^i, d_2^i, …, d_{ri}^i>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στο δένδρο δεινοσαύρων του κοπαδιού <$SpreadId_i$> (όπου έχουμε υποθέσει πως αυτά είναι r_i στο πλήθος).

Τέλος, h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \le i \le k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

- Y <N>: Γεγονός τύπου Year. Το γεγονός αυτό σηματοδοτεί την αύξηση της ηλικίας όλων των δεινόσαυρων κατά <N> έτη. Οι δεινόσαυροι που αποκτούν ηλικία μεγαλύτερη του MAX_AGE πεθαίνουν, όπου MAX_AGE είναι μια σταθερά που παρέχεται από τη γραμμή εντολών κατά την εκκίνηση του προγράμματος. Η δύναμη κάθε δεινόσαυρου επίσης αυξάνει κατά μία ποσότητα που είναι συνάρτηση της τρέχουσας ηλικίας του κάθε δεινόσαυρου.

  Συγκεκριμένα, διατρέχουμε τη λίστα των δεινόσαυρων και αυξάνουμε την ηλικία του κάθε δεινόσαυρου κατά <N>. Αν ο δεινόσαυρος αποκτά ηλικία μεγαλύτερη του MAX_AGE διαγράφεται από το σύστημα. Επιπρόσθετα, αυξάνουμε τη δύναμη του κάθε δεινόσαυρου κατά

  (1 + (MAX_AGE-Age)/MAX_AGE) * 12 * <N>,

όπου Age είναι η τρέχουσα ηλικία του κάθε δεινόσαυρου.

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

Y DONE

SET OF DINOSAURS:

<DId_1, Fed_1, $SpieceId_1$, $SpreadId_1$, Age_1, $Strength_1$>

<DId_2, Fed_2, $SpieceId_2$, $SpreadId_2$, Age_2, $Strength_2$>

…

<DId_h, Fed_h, $SpieceId_h$, $SpreadId_h$, Age_h, $Strength_h$>

όπου h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \leq i \leq k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

- P: Γεγονός τύπου Print που σηματοδοτεί την εκτύπωση των δομών δεδομένων του συστήματος. Συγκεκριμένα, για κάθε μία από τις διατροφικές κατηγορίες δεινοσαύρων, θα πρέπει να εκτυπώνεται το δένδρο ειδών δεινόσαυρων και το δένδρο κοπαδιών δεινόσαυρων της κατηγορίας αυτής. Επίσης, θα πρέπει να εκτυπώνονται τα στοιχεία του πίνακα κατακερματισμού δεινοσαύρων. Συγκεκριμένα, οι πληροφορίες που τυπώνονται είναι οι εξής:

P DONE

Eater[0]

SpieceTREE = <SpieceId1>,

List of Spiece <SpieceId1> = <d_1^1, d_2^1, …, d_{r1}^1>

< SpieceId2>,

List of Spiece <SpieceId2> = <d_1^2, d_2^2, …, d_{r2}^2>

…

…

<SpieceIdk>,

List of Spiece <SpieceIdk> = <d_1^k, d_2^k, …, d_{rk}^k>

SpreadTree = <SpreadId1>,

Tree of Spread <SpreadId1> = <d'^1_1, d'^1_2, …, d'^1_{m1}>

< SpreadId2>,

Tree of Spread <SpreadId2> = <d'^2_1, d'^2_2, …, d'^2_{m2}>

…

…

<SpreadIdl>,

Tree of Spread <SpreadIdl> = <d'^l_1, d'^l_2, …, d'^l_{ml}>

Eater[1]

<τυπώνεται παρόμοια πληροφορία όπως και για την κατηγορία Eater[0]>

<η τύπωση της πληροφορίας πρέπει να ακολουθεί παρόμοιο format

όπως εκείνο της πληροφορίας για την κατηγορία Eater[0]>

SET OF DINOSAURS:

<DId_1, Fed_1, $SpieceId_1$, $SpreadId_1$, Age_1, $Strength_1$>

<DId_2, Fed_2, $SpieceId_2$, $SpreadId_2$, Age_2, $Strength_2$>

…

<DId_h, Fed_h, $SpieceId_h$, $SpreadId_h$, Age_h, $Strength_h$>

όπου k, l είναι το πλήθος στοιχείων στα δένδρα ειδών δεινοσαύρων και κοπαδιών, αντίστοιχα, της διατροφικής κατηγορίας Eater[0] και <d_1^i, d_2^i, …, d_{ri}^i>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στη λίστα δεινοσαύρων του είδους <$SpieceId_i$> (όπου έχουμε υποθέσει πως αυτά είναι r_i στο πλήθος). Επίσης, <d'^i_1, d'^i_2, …, d'^i_{mi}>, $1 \le i \le k$, είναι τα αναγνωριστικά των δεινόσαυρων που είναι αποθηκευμένα στο δένδρο δεινοσαύρων του κοπαδιού <$SpreadId_i$> (όπου έχουμε υποθέσει πως αυτά είναι m_i στο πλήθος).

Τέλος, h είναι το συνολικό πλήθος των στοιχείων στις αλυσίδες του πίνακα κατακερματισμού των δεινοσαύρων και <DId_i, Fed_i, $Spiece_i$, $Spread_i$, Age_i, $Strength_i$>, $1 \le i \le k$, είναι το αναγνωριστικό, η διατροφική συνήθεια, το είδος, το κοπάδι, η ηλικία και η δύναμη του δεινόσαυρου που αντιστοιχεί στη i-οστή εγγραφή στην ακολουθία των εγγραφών όλων των αλυσίδων (θεωρώντας ότι εξετάζουμε την κάθε αλυσίδα με τη σειρά ξεκινώντας από την DIN[0] και καταλήγοντας στην DIN[M-1]).

Το πρόγραμμα που θα δημιουργηθεί πρέπει να διαβάζει το αρχείο εισόδου και να εκτελεί με τη σειρά όλα τα γεγονότα που περιγράφονται σε αυτό.

BONUS 1 (20 %)

Αντικαταστήστε το ταξινομημένο δυαδικό δένδρο ενός συγκεκριμένου κοπαδιού με ένα νηματικό δένδρο (το οποίο και πάλι θα πρέπει να είναι ταξινομημένο ως προς την ενδοδιατεταγμένη όπως ήταν αρχικά). Πληροφορίες για νηματικά δένδρα που είναι ταξινομημένα ως προς την ενδοδιατεταγμένη διάσχιση, όπως ζητείται εδώ (και όχι ως προς την προδιατεταγμένη διάσχιση που μελετήσατε στο 3° σετ ασκήσεων) παρουσιάζονται στην ιστοσελίδα του μαθήματος.

BONUS 2 (20 %)

Αντικαταστήστε το ταξινομημένο δυαδικό δένδρο ειδών δεινοσαύρων (το οποίο έχει κόμβο φρουρό) με ένα AVL δένδρο (με κόμβο φρουρό). Μπορείτε να αγνοήσετε τις διαγραφές που πρέπει να γίνονται στο δένδρο αυτό (δηλαδή να μην τις κάνετε, αλλάζοντας απλά το δείκτη SpieceDp της εκάστοτε εγγραφής που πρέπει να διαγραφεί με την τιμή NULL). Αυτό θα σας επιτρέψει να αποφύγεται την υλοποίηση της AVLDelete() που είναι μεγαλύτερης δυσκολίας.

Δομές Δεδομένων

Στο Σχήμα 4 παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της εργασίας.

```
typedef struct Dinosaur {
        int Did;
        int Fed;
```

```
        int SpieceId;
        int SpreadId;
        int Age;
        int Strength;
        struct Dinosaur *next;
} DINOSAUR;

// στοιχεία λίστας δεινοσαύρων ενός συγκεκριμένου είδους
typedef struct SpieceDin {
        int Did;
        struct SpieceDin *next;
} SPIECEDIN;

// κόμβοι δένδρου ειδών δεινοσαύρων
typedef struct SpieceNode {
        int SpieceId;
        struct SpieceDin *SpieceDp;
        struct SpieceNode *p;
        struct SpieceNode *lc;
        struct SpieceNode  *rc;
} SPIECENODE;


// κόμβοι δένδρου δεινοσαύρων ενός συγκεκριμένου κοπαδιού
typedef struct SpreadDin {
        DINOSAUR *d;
        struct SpreadDin *lc;
        struct SpreadDin *rc;
} SPREADDIN;

// κόμβοι δένδρου κοπαδιών
typedef struct SpreadNode {
        int SpreadId;
        struct SpreadDin *SpreadDp;
        struct SpreadNode *p;
        struct SpreadNode *lc;
        struct SpreadNode *rc;
} SPREADNODE;


// στοιχεία πίνακα Eater
typedef struct Feed {
        SPIECENODE *SpieceR;
        SPREADNODE *SpreadR;
} FEED;

// πίνακας Eater
FEED Eater[2];

// πίνακας κατακερματισμού
DINOSAUR DIN[M];
```

Οδηγίες και Συμβουλές για την Ομαλή Διεκπεραίωση της Εργασίας

Η εργασία θα πρέπει να πραγματοποιηθεί σε βήματα. Ο κάθε φοιτητής είναι υπεύθυνος να αποφασίσει ποια βήματα ταιριάζουν στον τρόπο εργασίας του. Στη συνέχεια, παρατίθεται μια δυνητική ακολουθία βημάτων που θα μπορούσε να ακολουθηθεί για την ομαλή διεκπεραίωση της εργασίας.

Βήμα 1: Αντιγράψτε τον κώδικα του Βήματος 5 της πρώτης φάσης και τροποποιήστε κατάλληλα τη λίστα που υλοποιήσατε εκεί προκειμένου να υλοποιήσετε το Ευριστικό Μετακίνησης στην Αρχή. Όταν σιγουρευτείτε πως το Ευριστικό εκτελείται σωστά, συνεχίστε με το επόμενο βήμα.

Βήμα 2: Ξεκινήστε (σε ένα νέο αρχείο) υλοποιώντας ένα μέρος του γεγονότος C (Create). Για την υλοποίηση του γεγονότος αυτού πρέπει να υλοποιήσετε το δένδρο ειδών δεινοσαύρων. Το δένδρο αυτό είναι διπλά-συνδεδεμένο και ταξινομημένο, ως προς το SpieceId, με κόμβο φρουρό. Στη φάση αυτή, αφήστε το πεδίο SpieceDp ίσο με NULL.

Φτιάξτε διαδικασίες DBSG-TreeInsert, DBSG-TreeDelete και DBSG-TreeLookUp για ένα διπλά-συνδεδεμένο, ταξινομημένο δυαδικό δένδρο με κόμβο φρουρό προκειμένου να πραγματοποιείτε εισαγωγές, διαγραφές και αναζητήσεις, αντίστοιχα, στο δένδρο αυτό. Τα στοιχεία του δένδρου αυτού είναι τύπου SpieceNode (όπου στη φάση αυτή, το πεδίο SpieceDp, αντίστοιχα, είναι ίσο με NULL). Δημιουργήστε μια διαδικασία DBSG-TreePrint() για την εκτύπωση των στοιχείων του δένδρου η οποία θα σας βοηθήσει να ελέγξετε την ορθότητα των διαδικασιών που περιγράφονται παραπάνω. Δημιουργήστε τον παραπάνω κώδικα σε ένα δικό σας αρχείο και φτιάξτε μια δική σας main() για να ελέγξετε ότι ο κώδικας που υλοποιεί τις λειτουργίες του δένδρου λειτουργεί σωστά. Όταν είστε σίγουροι για αυτό συνεχίστε στο επόμενο βήμα.

Βήμα 3: Συνεχίστε με την υλοποίηση του γεγονότος C, δημιουργώντας τον πίνακα Eater, κάθε στοιχείο του οποίου δεικτοδοτεί ένα δένδρο ειδών δεινοσαύρων (μην ασχοληθείτε στην παρούσα φάση με τα υπόλοιπα πεδία). Δημιουργήστε ένα αντίγραφο του βασικού αρχείου (εκείνου που περιέχει τη συνάρτηση main() που σας παρείχαν οι βοηθοί) και βάλτε σε σχόλια τον κώδικα που αναφέρεται σε γεγονότα άλλα από το C. Στο βήμα αυτό θα δουλέψετε μόνο με γεγονότα τύπου C. Απομονώστε στα test_files που σας παρείχαν οι βοηθοί τα γεγονότα τύπου C και εκτελέστε το πρόγραμμά σας μόνο για τέτοια γεγονότα. Το γεγονός τύπου C έχει βέβαια μόνο μερικά υλοποιηθεί αφού δεν υπάρχουν ακόμη οι λίστες δεινοσαύρων συγκεκριμένων ειδών, τα δένδρα δεινοσαύρων συγκεκριμένων κοπαδιών, ούτε το δένδρο κοπαδιών ή ο πίνακας κατακερματισμού δεινοσαύρων. Έτσι, στο βήμα αυτό απλά ελέγχετε αν τα στοιχεία του δένδρου ειδών κάθε διατροφικής κατηγορίας ενημερώνονται σωστά. Γεγονότα C που αναφέρονται στην προσθήκη νέων δεινόσαυρων σε κάποιο ήδη υπάρχον είδος δεν προκαλούν στη φάση αυτή καμία ενέργεια.

Βήμα 4: Συνεχίστε με την υλοποίηση του γεγονότος C, συγχωνεύοντας τον κώδικα που γράψατε στο Βήμα 1 παραπάνω (λίστες που υλοποιούν το Ευριστικό Μετακίνησης στην Αρχή) και τον κώδικα του Βήματος 3 ώστε να υλοποιήσετε τις λίστες δεινοσαύρων ενός συγκεκριμένου είδους.

Τελειώστε με το βήμα αυτό, ενημερώνοντας κατάλληλα το SpieceDp πεδίο κάθε στοιχείου του δένδρου ειδών μιας διατροφικής κατηγορίας ώστε να δείχνει στη ρίζα του δένδρου δεινοσαύρων του συγκεκριμένου είδους στο οποίο αναφέρεται. Με τον τρόπο αυτό έχετε φτιάξει ένα δένδρο από λίστες (αφού κάθε στοιχείο του δένδρου ειδών αποθηκεύει ένα δείκτη στο πρώτο στοιχείο της λίστας δεινοσαύρων που ανήκουν στο συγκεκριμένο είδος).

Βήμα 5: Σε ένα νέο αρχείο γράψτε κώδικα που να υλοποιεί το δένδρο κοπαδιών δεινοσαύρων. Το δένδρο αυτό είναι διπλά-συνδεδεμένο, προδιατεταγμένα-ταξινομημένο ως προς το πεδίο αναγνωριστικό. Φτιάξτε διαδικασίες DPS-TreeInsert, DPS-TreeDelete και DPS-TreeLookUp για ένα διπλά-συνδεδεμένο, προδιατεταγμένα-ταξινομημένο, προκειμένου να πραγματοποιείτε εισαγωγές, διαγραφές και αναζητήσεις, αντίστοιχα, στο δένδρο αυτό. Τα στοιχεία του δένδρου αυτού μπορούν να είναι τύπου SpreadNode (όπου στη φάση αυτή, το πεδίο SpreadDp είναι προσωρινά ίσο με NULL). Δημιουργήστε μια διαδικασία DPS-TreePrint() για την εκτύπωση των στοιχείων του δένδρου η οποία θα σας βοηθήσει να ελέγξετε την ορθότητα των διαδικασιών που περιγράφονται παραπάνω. Δημιουργήστε τον παραπάνω κώδικα σε ένα δικό σας αρχείο και φτιάξτε μια δική σας main() για να ελέγξετε ότι ο κώδικας που υλοποιεί τις λειτουργίες του δένδρου λειτουργεί σωστά. Όταν είστε σίγουροι για αυτό συνεχίστε στο επόμενο βήμα.

Βήμα 6: Σε ένα νέο αρχείο αντιγράψτε τον κώδικα που φτιάξατε στο Βήμα 2 και τροποποιήστε τον κατάλληλα ώστε να υλοποιεί ταξινομημένα, ως προς το Strength, δένδρα χωρίς κόμβο φρουρό, τα οποία δεν είναι διπλά-συνδεδεμένα. Οι κόμβοι των δένδρων αυτών είναι τύπου SpreadDin. Προσέξτε ότι στο δένδρο ενός συγκεκριμένου κοπαδιού θα πρέπει να μπορείτε να εισάγετε ένα σύνολο από στοιχεία. Σκεφτείτε ποιος είναι ο πιο αποδοτικός τρόπος για να επιτευχθεί αυτό.

Στο βήμα αυτό, τροποποιήστε το struct SpreadDin και αντί να σώζεται στο πεδίο d έναν δείκτη σε ένα struct Dinosaur μπορείτε να αποθηκεύεται εκεί απλά το αναγνωριστικό του δεινόσαυρου. Αυτό θα σήμαινε βέβαια πως όλες τις άλλες πληροφορίες για τον εκάστοτε δεινόσαυρο τις αγνοείτε στην παρούσα φάση.

Φτιάξτε διαδικασίες BS-TreeInsert, BS-TreeDelete και BS-TreeLookUp για ένα απλά-συνδεδεμένο, ταξινομημένο δυαδικό δένδρο χωρίς κόμβο φρουρό προκειμένου να πραγματοποιείτε εισαγωγές, διαγραφές και αναζητήσεις, αντίστοιχα, στο δένδρο αυτό. Δημιουργήστε μια διαδικασία BS-TreePrint() για την εκτύπωση των στοιχείων του δένδρου η οποία θα σας βοηθήσει να ελέγξετε την ορθότητα των διαδικασιών που περιγράφονται παραπάνω. Φτιάξτε μια δική σας main() για να ελέγξετε ότι ο κώδικας που υλοποιεί τις λειτουργίες του δένδρου λειτουργεί σωστά. Όταν είστε σίγουροι για αυτό συνεχίστε στο επόμενο βήμα.

Βήμα 7: Συνενώστε τους κώδικες των Βημάτων 5 και 6 για να τελειώσετε την υλοποίηση των δένδρων κοπαδιών.

Ενημερώστε κατάλληλα το SpreadDp πεδίο κάθε στοιχείου του δένδρου κοπαδιών μιας διατροφικής κατηγορίας ώστε να δείχνει στη ρίζα του δένδρου δεινοσαύρων του κοπαδιού στο οποίο αναφέρεται. Με τον τρόπο αυτό έχετε φτιάξει ένα δένδρο από δένδρα (αφού κάθε στοιχείο του δένδρου κοπαδιών αποθηκεύει ένα δείκτη στη ρίζα ενός δένδρου δεινοσαύρων που ανήκουν στο συγκεκριμένο κοπάδι).

Δημιουργήστε ένα μέρος της Print(), η οποία θα τυπώνει μόνο την εξής πληροφορία:

P DONE
Eater[0]
SpieceLIST = <SpieceId1>,
List of Spiece <SpieceId1> = <d_1^1, d_2^1, …, d_{r1}^1>
< SpieceId2>,
List of Spiece <SpieceId2> = <d_1^2, d_2^2, …, d_{r2}^2>
…
…
<SpieceIdk>,
List of Spiece <SpieceIdk> = <d_1^k, d_2^k, …, d_{rk}^k>

SpreadList = <SpreadId1>,
List of Spread <SpreadId1> = <d'^1_1, d'^1_2, …, d'^1_{m1}>
< SpreadId2>,
List of Spread <SpreadId2> = <d'^2_1, d'^2_2, …, d'^2_{m2}>
…
…
<SpreadIdl>,
List of Spread <SpreadIdl> = <d'^l_1, d'^l_2, …, d'^l_{ml}>
Eater[1]
<τυπώνεται παρόμοια πληροφορία όπως και για την κατηγορία Eater[0]>
<η τύπωση της πληροφορίας πρέπει να ακολουθεί παρόμοιο format
όπως εκείνο της πληροφορίας για την κατηγορία Eater[0]>

Βήμα 8: Στο βήμα αυτό θα υλοποιήσετε τον πίνακα κατακερματισμού δεινοσαύρων που επιλύει συγκρούσεις με τη μέθοδο των αλυσίδων, όπου οι αλυσίδες είναι ταξινομημένες ως προς το αναγνωριστικό (Did) των δεινοσαύρων. Κάθε στοιχείο του πίνακα αυτού περιέχει έναν δείκτη στο πρώτο στοιχείο μιας αλυσίδας που περιέχει εκείνα τα struct Dinosaur για τα οποία το id τους έχει τιμή κατακερματισμού ίση με i. Γράψτε κώδικα που θα υλοποιεί τον πίνακα αυτό σε ένα εντελώς νέο αρχείο.

Φτιάξτε διαδικασίες HT_Insert, HT_Delete και HT_LookUp οι οποίες πραγματοποιούν εισαγωγές, διαγραφές και αναζητήσεις, αντίστοιχα, στα στοιχεία των αλυσίδων του πίνακα κατακερματισμού. Δημιουργήστε μια διαδικασία HT_Print() για την εκτύπωση των στοιχείων των αλυσίδων του πίνακα η οποία θα σας βοηθήσει να ελέγξετε την ορθότητα των διαδικασιών που περιγράφονται παραπάνω. Τροποποιήστε κατάλληλα τη main() του Βήματος 1 για να ελέγξετε ότι ο κώδικας που υλοποιεί τις λειτουργίες του πίνακα κατακερματισμού λειτουργεί σωστά. Όταν είστε σίγουροι για αυτό συνεχίστε στο επόμενο βήμα.

Βήμα 9: Δοκιμάστε να τρέξετε γεγονότα όλων των ειδών πάνω στον πίνακα κατακερματισμού δεινοσαύρων. Χρησιμοποιήστε επομένως, αυτούσια τα test_files και τη main που θα σας παρέχουν οι βοηθοί και ελέγξετε πως

εκτελούνται σωστά (αν και με μη-πλήρη τρόπο αφού λείπουν οι περισσότερες δομές της εργασίας από τον κώδικα του βήματος αυτού) όλων των ειδών τα γεγονότα στη λίστα δεινοσαύρων.

Βήμα 10: Συνδυάστε τους κώδικες που δημιουργήσατε στο Βήματα 8 και 6 ώστε να πάρετε μια πρώτη έκδοση της εργασίας σας με όλες τις δομές που ζητούνται υλοποιημένες. Ολοκληρώστε την υλοποίηση των γεγονότων C, B, Y και P.

Βήμα 11: Υλοποιήστε το γεγονός τύπου S. Βεβαιωθείτε πως και αυτό το γεγονός εκτελείται σωστά.

Βήμα 12: Υλοποιήστε το γεγονός τύπου W. Αυτό θα σας επιτρέψει να πειραματιστείτε με διαγραφές και άλλες πολύπλοκες λειτουργίες πάνω στις δομές.

Βήμα 13: Ελέγξετε την ορθότητα του κώδικα που δημιουργήσατε εκτελώντας τον κώδικά σας σε όλα τα αρχεία εκτέλεσης που θα σας παρέχουν οι βοηθοί του μαθήματος. Επιπρόσθετα, δημιουργήστε τα δικά σας αρχεία γεγονότων για να ελέγξετε με περισσότερη ακρίβεια την ορθότητα του κώδικά σας.