

## HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-2019

Διδάσκουσα: Παναγιώτα Φατούρου

Προγραμματιστική Εργασία - 2ο Μέρος

Ημερομηνία Παράδοσης: Παρασκευή, 21 Δεκεμβρίου 2018, ώρα 23:59

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



Κι απ' την θαυμάσια πανελλήνιαν εκστρατεία,  
την νικηφόρα, την περιλαμπρή,  
την περιάλλητη, την δοξασμένη  
ως άλλη δεν δοξάσθηκε καμιά,  
την απaráμιλλη: βγήκαμ' εμείς  
ελληνικός καινούριος κόσμος, μέγας.

Εμείς. Οι Αλεξανδρείς, οι Αντιοχείς,  
οι Σελεύκεις, κ' οι πολυάριθμοι  
επίλοιποι Έλληνες Αιγύπτου και Συρίας,  
κ' οι εν Μηδία, κ' οι εν Περσίδι, κι όσοι άλλοι.  
Με τες εκτεταμένες επικράτειες,  
με την ποικίλη δράσι των στοχαστικών προσαρμογών.  
Και την Κοινήν Ελληνική Λαλιά  
ως μέσα στην Βακτριανή την πήγαμεν, ως τους Ινδούς.

«Στα 200 π.Χ.»  
Κ.Π. Καβάφης

Φωτογραφία:

<http://www.historyofmacedonia.org/AncientMacedonia/AlexandertheGreat.html>

«Ο Μέγας Αλέξανδρος ήταν ο καταλύτης για την παγκοσμιοποίηση του Ελληνισμού. Ένας άνθρωπος που παρήγαγε πολιτισμικό «παράδειγμα» με διάρκεια ιστορικού βίου δεκαοχτώ περίπου αιώνων.

Στον κυρίως γεωγραφικό χώρο της ελληνικής παρουσίας - στις δύο ακτές του Αιγαίου και στο ενδιάμεσο Αρχιπέλαγος - γεννήθηκε, για πρώτη φορά στην ιστορία του ανθρώπου, η ανάγκη να επαληθεύεται η γνώση: Να μπορούν να ξεχωρίζουν οι άνθρωποι τη βέβαιη από την αβέβαιη γνώση, την αξιόπιστη από την αναξιόπιστη πληροφορία, εντύπωση, πιστοποίηση, - να διακρίνουν το σωστό από το λάθος, την αλήθεια από το ψέμα. Γεννήθηκε η κριτική σκέψη.»

Χρήστος Γιανναράς

## Γενική Περιγραφή Εργασίας

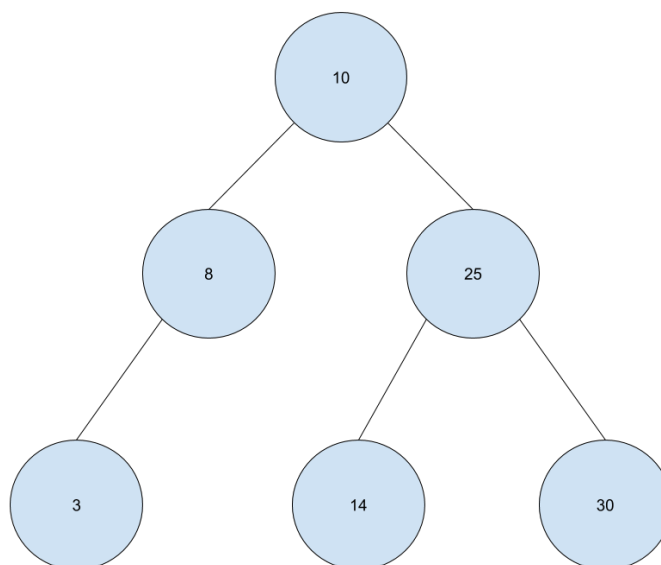
Στη εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που να προσομοιώνει τη μάχη στην Περσίδα Πύλη, η οποία υπήρξε μία από τις τελευταίες και πιο καταλυτικές μάχες που έδωσε ο στρατός του Μεγάλου Αλεξάνδρου έναντι του σατράπη Αριοβαρζάνη το 330 ΠΚΕ, ανοίγοντας το δρόμο για την Περσέπολη, ένα από τα κέντρα της δυναστείας των Αχαιμενιδών.

## Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Ο στρατός του Μεγάλου Αλεξάνδρου αποτελείται από ένα σύνολο στρατιωτών και αλόγων. Οι στρατιώτες αποθηκεύονται σε ένα δένδρο δυαδικής αναζήτησης με όνομα **δένδρο στρατού του Μεγάλου Αλεξάνδρου**. Τα στοιχεία του δένδρου αυτού είναι **ταξινομημένα** με βάση το αναγνωριστικό των στρατιωτών. Ο κάθε κόμβος του δένδρου είναι μία εγγραφή τύπου *struct Army* με τα ακόλουθα πεδία:

- **id**: Αναγνωριστικό (τύπου *int*) που χαρακτηρίζει μοναδικά τον στρατιώτη.
- **rc**: Δείκτης (τύπου *struct Army*) στο δεξί παιδί του κόμβου.
- **lc**: Δείκτης (τύπου *struct Army*) στο αριστερό παιδί του κόμβου.

Στο Σχήμα 1 παρουσιάζεται το δένδρο στρατού του Μεγάλου Αλεξάνδρου.



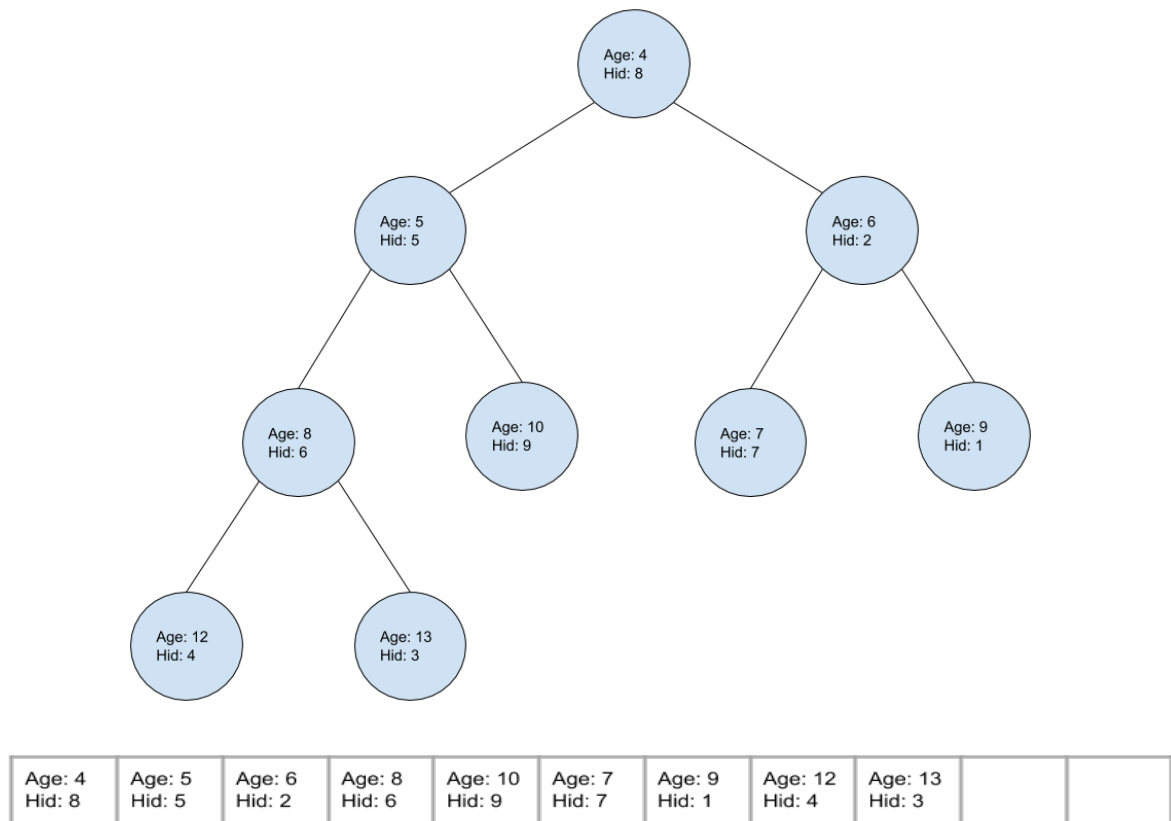
**Σχήμα 1:** Το δένδρο δυαδικής αναζήτησης “δένδρο στρατού” του Μεγάλου Αλεξάνδρου. Οι κόμβοι είναι ταξινομημένοι ως προς το αναγνωριστικό τους.

Τα άλογα του στρατού αποθηκεύονται σε μία ουρά προτεραιότητας, υλοποιημένη ως **σωρό**, με όνομα **ουρά προτεραιότητας αλόγων του Μεγάλου Αλεξάνδρου**. Ο κάθε κόμβος του πίνακα που υλοποιεί τον σωρό, είναι μία εγγραφή τύπου *struct Horse* με τα ακόλουθα πεδία:

- **id**: Αναγνωριστικό (τύπου *int*) που χαρακτηρίζει μοναδικά το άλογο.
- **age**: Η ηλικία του αλόγου (τύπου *int*)

Η προτεραιότητα στην ουρά αλόγων θα καθορίζεται με βάση το πεδίο **age** των αλόγων. Με άλλα

λόγια, τα άλογα με μικρότερη ηλικία θα πρέπει να έχουν μεγαλύτερη προτεραιότητα έναντι των γηραιότερων αλόγων. Στο Σχήμα 2 παρουσιάζεται η ουρά αλόγων του Μεγάλου Αλεξάνδρου.



**Σχήμα 2:** Η ουρά προτεραιότητας αλόγων του Μεγάλου Αλεξάνδρου. Η προτεραιότητα των κόμβων καθορίζεται από το πεδίο age. Στο πάνω σχήμα βλέπετε βοηθητικά το δένδρο του σωρού.

Συγκεκριμένα, η ουρά προτεραιότητας θα είναι μία εγγραφή τύπου *struct HorsePQ* με τα ακόλουθα πεδία:

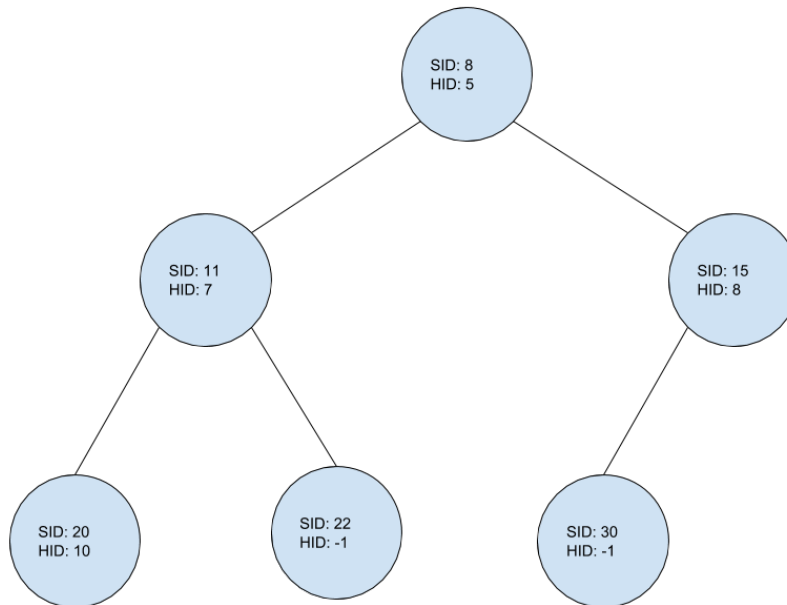
- **size:** Μετρητής (τύπου *int*) που υποδηλώνει πόσα στοιχεία έχει η ουρά την εκάστοτε χρονική στιγμή.
- **HorsePQ[]:** Ένας πίνακας (με στοιχεία τύπου *struct Horse*) που θα δεσμεύεται δυναμικά και θα περιέχει τα στοιχεία της ουράς προτεραιότητας.

Για την προετοιμασία του στρατού του Μεγάλου Αλεξάνδρου για τη μάχη στην Περσίδα Πύλη, οι στρατιώτες μαζί με τα άλογα αποθηκεύονται σε ένα **πλήρες συνδεδεμένο δένδρο** (μη ταξινομημένο), με όνομα **δένδρο μάχης Μεγάλου Αλεξάνδρου**. Επομένως, κάθε εισαγωγή ή διαγραφή που γίνεται σ' αυτό το δένδρο θα πρέπει να μην καταστρατηγεί την ιδιότητα ότι το δένδρο είναι πλήρες. Ο κάθε κόμβος του δένδρου είναι μία εγγραφή, τύπου *struct GA\_Battle*, που περιέχει τα ακόλουθα πεδία:

- **soldier\_id:** Αναγνωριστικό (τύπου *int*) που χαρακτηρίζει μοναδικά τον στρατιώτη.
- **horse\_id:** Αναγνωριστικό (τύπου *int*) που χαρακτηρίζει μοναδικά το άλογο (-1 αν ο στρατιώτης είναι πεζός).
- **rc:** Δείκτης (τύπου *struct GA\_Battle*) στο δεξί παιδί του κόμβου.

- **lc**: Δείκτης (τύπου *struct GA\_Battle*) στο αριστερό παιδί του κόμβου.

Στο Σχήμα 3, παρουσιάζεται το δένδρο μάχης τους Μεγάλου Αλεξάνδρου.



**Σχήμα 3:** Το πλήρες δένδρο μάχης του Μεγάλου Αλεξάνδρου.

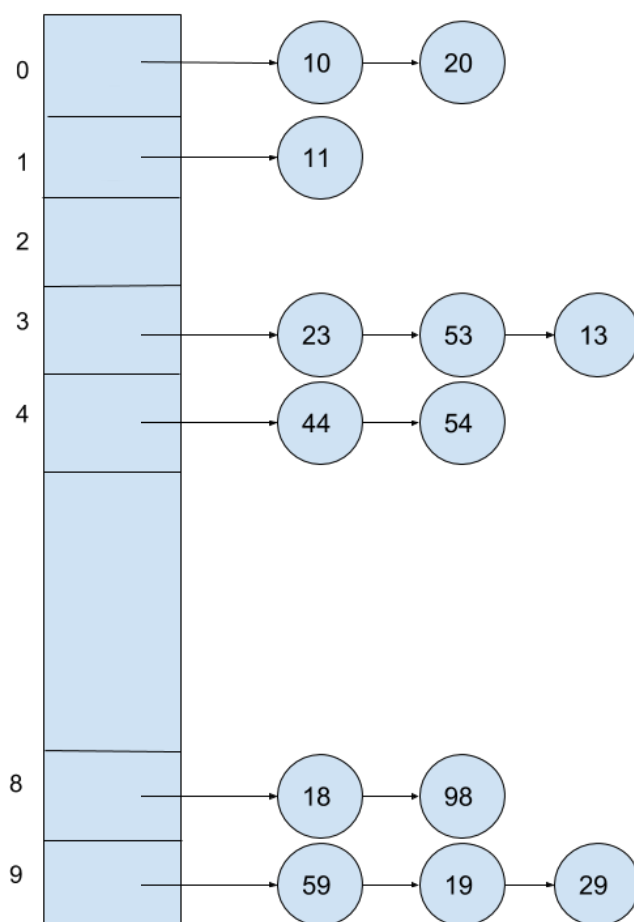
Συγκεκριμένα, το πλήρες δένδρο μάχης θα αναπαρίσταται από μία εγγραφή τύπου *struct GA\_Tree* με τα ακόλουθα πεδία:

- **size**: Μετρητής (τύπου *int*) που υποδηλώνει πόσα στοιχεία έχει το δένδρο την εκάστοτε χρονική στιγμή.
- **GA\_tree**: Ένας δείκτης (τύπου *struct GA\_Battle\**) που θα δείχνει στη ρίζα του δένδρου.

Έναντι στα στρατεύματα του Μεγάλου Αλεξάνδρου, βρίσκονταν τα στρατεύματα του Αριοβαρζάνη. Ο στρατός του Αριοβαρζάνη αποτελείται από πεζούς οπλίτες και κάθε στρατιώτης αποθηκεύεται με βάση το αναγνωριστικό του σε έναν πίνακα κατακερματισμού με όνομα **πίνακας κατακερματισμού μάχης του Αριοβαρζάνη**. Οι συγκρούσεις στον πίνακα κατακερματισμού θα πρέπει να επιλύονται με τη μέθοδο των ξεχωριστών αλυσίδων (*separate chaining*). Για την υλοποίηση της συνάρτησης κατακερματισμού, θα πρέπει να χρησιμοποιήσετε καθολικό κατακερματισμό (*universal hashing*). Για την υλοποίηση του καθολικού κατακερματισμού, παρέχεται ένας πίνακας *primes\_g[]* με πρώτους αριθμούς σε αύξουσα σειρά, το μέγιστο πλήθος στρατιωτών, μέσω της μεταβλητής *max\_soldiers\_g* και το μέγιστο αναγνωριστικό μεταξύ των στρατιωτών, μέσω της μεταβλητής *max\_soldiers\_id\_g*. Ο πίνακας και οι δύο αυτές μεταβλητές είναι *global* κι έχουν δηλωθεί στο αρχείο *battle.h*. Ο κάθε κόμβος της κάθε αλυσίδας που θα δημιουργείται στον πίνακα κατακερματισμού αποτελεί μία εγγραφή τύπου *struct AR\_Battle* που περιέχει τα ακόλουθα πεδία:

- **id**: Αναγνωριστικό (τύπου *int*) που χαρακτηρίζει μοναδικά τον κάθε στρατιώτη.
- **next**: Δείκτης (τύπου *struct AR\_Battle*) στον επόμενο κόμβο της αλυσίδας.

Στο Σχήμα 4 παρουσιάζεται ο πίνακας μάχης του Αριοβαρζάνη.



**Σχήμα 4:** Ο πίνακας κατακερματισμού “πίνακας μάχης” του Αριοβαρζάνη. Οι στρατιώτες εισάγονται στην αντίστοιχη αλυσίδα με βάση το αναγνωριστικό τους. Στο παράδειγμα, η συνάρτηση κατακερματισμού είναι:  $h(\text{key}) = \text{key} \bmod 10$ .

### 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), η πρώτη γραμμή του οποίου θα περιέχει μόνο έναν ακέραιο <H>, που υποδηλώνει ποιος είναι ο μέγιστος αριθμός αλόγων για το συγκεκριμένο σενάριο μάχης, ώστε να δεσμεύσετε δυναμικά τον κατάλληλο πίνακα που θα υλοποιεί την ουρά προτεραιότητας. Ακόμη, η δεύτερη και η τρίτη γραμμή θα περιέχουν επίσης μόνο έναν ακέραιο η καθεμία που θα υποδηλώνουν τον μέγιστο πλήθος στρατιωτών του Αριοβαρζάνη και το μέγιστο αναγνωριστικό αυτών αντίστοιχα, ώστε να υλοποιήσετε τον καθολικό κατακερματισμό ανάλογα. Στην συνέχεια το αρχείο εισόδου θα περιέχει γεγονότα των ακόλουθων μορφών:

### R <sid>

Γεγονός τύπου Register Alexander the Great soldier που υποδηλώνει την εισαγωγή ενός νέου στρατιώτη με αναγνωριστικό <sid> στα στρατεύματα του Μεγάλου Αλεξάνδρου. Κατά το γεγονός

αυτό, θα γίνεται εισαγωγή ενός νέου κόμβου τύπου *struct Army* στο δένδρο στρατού του Μεγάλου Αλεξάνδρου. Μετά από κάθε εισαγωγή, το δένδρο θα πρέπει να μένει ταξινομημένο. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
R <sid>
GA soldiers = <sid1>, <sid2>, <sid3>, ..., <sidn>
DONE
```

Όπου  $n$  είναι ο αριθμός των κόμβων στο δένδρο στρατού του Μεγάλου Αλεξάνδρου και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $\langle \text{sid}_i \rangle$  είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου σύμφωνα με την ενδοδιατεταγμένη διάσχιση..

### H <hid> <age>

Γεγονός τύπου register Alexander the Great Horse που υποδηλώνει την εισαγωγή ενός νέου αλόγου στα στρατεύματα του Μεγάλου Αλεξάνδρου με αναγνωριστικό <hid> και ηλικία <age>. Κατά το γεγονός αυτό, θα γίνεται εισαγωγή ενός νέου κόμβου τύπου *struct Horse* στην ουρά προτεραιότητας αλόγων του Μεγάλου Αλεξάνδρου. Μετά από κάθε εισαγωγή η δομή θα πρέπει να εξακολουθεί να είναι ουρά προτεραιότητας. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
H <hid> <age>
Level 0: <hid0, age0>
Level 1: <hid1, age1>, <hid2, age2>
...
Level N: <hidN,0, ageN,0>, <hidN,1, ageN,1>, ..., <hidN,n, ageN,n>
DONE
```

όπου  $N$  είναι το ύψος του δένδρου που αντιστοιχεί στην ουρά,  $n$  είναι ο αριθμός των κόμβων στην ουρά προτεραιότητας αλόγων του Μεγάλου Αλεξάνδρου και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $\langle \text{hid}_i \rangle$  και  $\langle \text{age}_i \rangle$  είναι το αναγνωριστικό και η ηλικία του αλόγου που αντιστοιχεί στον  $i$ -οστό κόμβο της ουράς προτεραιότητας.

### A <aid>

Γεγονός τύπου register satrapy soldier που υποδηλώνει την εισαγωγή ενός νέου στρατιώτη με αναγνωριστικό <aid> στο στρατό του Αριοβαρζάνη. Κατά το γεγονός αυτό, θα γίνεται εισαγωγή ενός νέου κόμβου τύπου *struct AR\_Battle* στον πίνακα κατακερματισμού μάχης του Αριοβαρζάνη. Συγκεκριμένα, το αναγνωριστικό του νέου στρατιώτη θα πρέπει να δίνεται ως είσοδος στην συνάρτηση κατακερματισμού και στη συνέχεια ο νέας κόμβος θα εισάγεται στην αντίστοιχη αλυσίδα του πίνακα. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```

A <aid>
Index 0: <aid0,1>, <aid0,2>, ..., <aid0,N0>
Index 1: <aid1,1>, <aid1,2>, ..., <aid1,N1>
...
Index n: <aidn,1>, <aidn,2>, ..., <aidn,Nn>
DONE

```

όπου  $n$  είναι το πλήθος των θέσεων του πίνακα κατακερματισμού του Αριοβαρζάνη και για κάθε  $j$ ,  $0 \leq j \leq n-1$ ,  $N_j$  είναι το πλήθος των κόμβων στην αλυσίδα της θέσης  $j$  του πίνακα κατακερματισμού, όπως και για κάθε  $i$ ,  $1 \leq i \leq N_j$ ,  $\langle \text{aid}_{j,i} \rangle$  είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον  $i$ -οστό κόμβο της αλυσίδας που βρίσκεται στην θέση  $j$  του πίνακα κατακερματισμού.

### P

Γεγονός τύπου *prepare for battle* για τα στρατεύματα του Μεγάλου Αλεξάνδρου. Κατά το γεγονός αυτό, θα πρέπει να διασχίσετε το δένδρο στρατού του Μεγάλου Αλεξάνδρου σύμφωνα με την ενδοδιατεταγμένη διάσχιση. Κάθε στρατιώτης, πριν προετοιμαστεί για μάχη, θα πρέπει πρώτα να δεσμεύει ένα διαθέσιμο άλογο από την ουρά αλόγων. Για τον σκοπό αυτό, κάθε στρατιώτης θα διαγράφει το πρώτο διαθέσιμο άλογο από την ουρά αλόγων σύμφωνα με την προτεραιότητα που πρέπει να τηρείται, δηλαδή το άλογο εκείνο με τη μικρότερη ηλικία. Αν δεν υπάρχουν άλλα διαθέσιμα άλογα, ο στρατιώτης θα εισάγεται στο δένδρο μάχης του Μ. Αλεξάνδρου ως πεζός. Στην συνέχεια, θα πρέπει να εισάγετε έναν νέο κόμβο τύπου *struct GA\_Battle* στο δένδρο μάχης του Μεγάλου Αλεξάνδρου. Ο νέος κόμβος του δένδρου μάχης θα έχει ως τιμή στο πεδίο  $\langle \text{soldier\_id} \rangle$  το αναγνωριστικό του στρατιώτη. Το πεδίο  $\langle \text{horse\_id} \rangle$  θα έχει ως τιμή το αναγνωριστικό του αλόγου που δέσμευσε ο στρατιώτης, ή την τιμή  $-1$ , σε περίπτωση που ο στρατιώτης είναι πεζός. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```

P
GA battle = <sid1, hid1>, <sid2, hid2>, <sid3, hid3>, ..., <sidn, hidn>
DONE

```

Όπου  $n$  είναι ο αριθμός των κόμβων στο δένδρο μάχης του Μεγάλου Αλεξάνδρου και για κάθε  $i$ ,  $1 \leq i \leq n$ ,  $\langle \text{sid}_i \rangle$  είναι το αναγνωριστικό του στρατιώτη και  $\langle \text{hid}_i \rangle$  είναι το αναγνωριστικό του αλόγου του (αν ο στρατιώτης είναι ιππέας, διαφορετικά το  $\langle \text{hid}_i \rangle$  έχει την τιμή  $-1$ ), που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου μάχης σύμφωνα με την ενδοδιατεταγμένη διάσχιση.

### T <X>

Γεγονός τύπου *ambush*, το οποίο προσομοιώνει την παγίδα που είχε στήσει ο Αριοβαρζάνης στον Αλέξανδρο έξω από την Περσίδα Πύλη και προκάλεσε σημαντικές απώλειες στον στρατό του. Κατά το γεγονός αυτό, θα πρέπει να διασχίσετε το δένδρο μάχης του Μεγάλου Αλεξάνδρου σύμφωνα με κάποια από τις αναδρομικές διασχίσεις και να διαγράφετε έναν στρατιώτη κάθε  $\langle X \rangle$  στρατιώτες. Μετά από κάθε διαγραφή το δένδρο μάχης θα πρέπει να παραμένει πλήρες. Επομένως, κάθε φορά που διαγράφεται ένας κόμβος θα πρέπει να εντοπίζετε το δεξιότερο φύλλο στο πλήρες δένδρο και να αντικαθιστάτε τον κόμβο προς διαγραφή με αυτό. Μετά το πέρας της



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

**T <X>**

**GA battle = <sid<sub>1</sub>, hid<sub>1</sub>>, <sid<sub>2</sub>, hid<sub>2</sub>>, <sid<sub>3</sub>, hid<sub>3</sub>>, ..., <sid<sub>n</sub>, hid<sub>n</sub>>**

**DONE**

Όπου  $n$  είναι ο αριθμός των κόμβων στο δένδρο μάχης του Μεγάλου Αλεξάνδρου και για κάθε  $i$ ,  $1 \leq i \leq n$ , <sid <sub>$i$</sub> > είναι τα αναγνωριστικό του στρατιώτη και <hid <sub>$i$</sub> > είναι το αναγνωριστικό του αλόγου του (αν ο στρατιώτης είναι ιππέας, διαφορετικά το <hid <sub>$i$</sub> > έχει την τιμή -1), που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου μάχης σύμφωνα με την ενδοδιατεγμένη διάσχιση.

**K**

Γεγονός τύπου soldiers' death στον στρατό του Αριοβαρζάνη, το οποίο προσομοιώνει την ανατροπή του στρατού του Μεγάλου Αλεξάνδρου, όταν οδηγώντας τον στρατό του πίσω από τον εχθρό τον αιφνιδίασε κι εν τέλει κέρδισε την μάχη. Κατά το γεγονός αυτό, θα διασχίσετε το δένδρο μάχης του Μεγάλου Αλεξάνδρου και για κάθε στρατιώτη θα χρησιμοποιείτε το id του σαν είσοδο στην συνάρτηση κατακερματισμού για τον πίνακα μάχης του Αριοβαρζάνη. Στην συνέχεια, θα διαγράφετε τους τρεις πρώτους στρατιώτες της αντίστοιχης αλυσίδας στον πίνακα μάχης. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

**K**

**Index 0: <aid<sub>0,1</sub>>, <aid<sub>0,2</sub>>, ..., <aid<sub>0,N<sub>0</sub></sub>>**

**Index 1: <aid<sub>1,1</sub>>, <aid<sub>1,2</sub>>, ..., <aid<sub>1,N<sub>1</sub></sub>>**

**...**

**Index n: <aid<sub>n,1</sub>>, <aid<sub>n,2</sub>>, ..., <aid<sub>N,n</sub>>**

**DONE**

όπου  $n$  είναι το πλήθος των θέσεων του πίνακα κατακερματισμού του Αριοβαρζάνη και για κάθε  $j$ ,  $0 \leq j \leq n-1$ ,  $N_j$  είναι το πλήθος των κόμβων στην αλυσίδα που δεικτοδοτείται από τη θέση  $j$  του πίνακα κατακερματισμού, ενώ για κάθε  $1 \leq i \leq N_j$ , <aid <sub>$j,i$</sub> > είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον  $i$ -οστό κόμβο της αλυσίδας που δεικτοδοτείται από τη θέση  $j$  του πίνακα κατακερματισμού.

**D**

Γεγονός που δηλώνει το θάνατο του Μεγάλου Αλεξάνδρου και τον διαμοιρασμό του στρατού του στους εξής 5 διαδόχους: **Πτολεμαίος, Δημήτριος ο Πολιορκητής, Κάσσανδρος, Λυσίμαχος και Σέλευκος**. Κατά το γεγονός αυτό, θα διαμοιράσετε (split) το δένδρο στρατού του Μεγάλου Αλεξάνδρου σε 5 νέα δένδρα δυαδικής αναζήτησης (δηλαδή τα δένδρα αυτά είναι ταξινομημένα), ένα για κάθε στρατηγό. Συγκεκριμένα, οι στρατιώτες με id στο διάστημα **[0, 500)**, θα ανατίθενται στο δένδρο του πρώτου στρατηγού, αυτοί με id στο διάστημα **[500, 1000)** στο δένδρο του δεύτερου στρατηγού κ.ο.κ. Η πολυπλοκότητα του γεγονότος αυτού θα πρέπει να είναι **O(h)**, όπου  $h$  είναι το ύψος του δένδρου στρατού του Μεγάλου Αλεξάνδρου. Για να το πετύχετε αυτό,



ξεκινώντας από την ρίζα θα πρέπει να “κόβετε” το αρχικό δένδρο στα σημεία που χρειάζεται και να μεταφέρετε ολόκληρα εκείνα τα υποδένδρα που περιέχουν τα «κατάλληλα» κλειδιά στο δένδρο του αντίστοιχου στρατηγού.

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

**D**

**General 0 = <sid<sub>0,1</sub>>, <sid<sub>0,2</sub>>, <sid<sub>0,3</sub>>, ..., <sid<sub>0,n0</sub>>**

**General 1 = <sid<sub>1,1</sub>>, <sid<sub>1,2</sub>>, <sid<sub>1,3</sub>>, ..., <sid<sub>1,n1</sub>>**

**...**

**General 4 = <sid<sub>4,1</sub>>, <sid<sub>4,2</sub>>, <sid<sub>4,3</sub>>, ..., <sid<sub>4,n4</sub>>**

**DONE**

Όπου για κάθε  $0 \leq j \leq 4$ ,  $n_j$  είναι ο αριθμός των κόμβων στο δένδρο στρατού του κάθε στρατηγού και για κάθε  $1 \leq i \leq n_j$ ,  $\langle \text{sid}_{j,i} \rangle$  είναι το αναγνωριστικό του στρατιώτη που αντιστοιχεί στον  $i$ -οστό κόμβο του δένδρου αυτού σύμφωνα με την ενδοδιατεταγμένη διάταξη.

**X**

Γεγονός το οποίο σηματοδοτεί την εκτύπωση του δένδρου στρατού του Μεγάλου Αλεξάνδρου. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

**X**

**GA soldiers = <sid<sub>1</sub>>, <sid<sub>2</sub>>, <sid<sub>3</sub>>, ..., <sid<sub>n</sub>>**

**DONE**

όπως περιγράφηκε στο γεγονός **R**.

**Y**

Γεγονός το οποίο σηματοδοτεί την εκτύπωση της ουράς αλόγων του Μεγάλου Αλεξάνδρου. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

**Y**

**Level 0: <hid<sub>0</sub>, age<sub>0</sub>>**

**Level 1: <hid<sub>1</sub>, age<sub>1</sub>>, <hid<sub>2</sub>, age<sub>2</sub>>**

**...**

**Level N: <hid<sub>N,0</sub>, age<sub>N,0</sub>>, <hid<sub>N,1</sub>, age<sub>N,1</sub>>, ..., <hid<sub>N,n</sub>, age<sub>N,n</sub>>**

**DONE**

όπως περιγράφηκε στο γεγονός **H**.

**Z**

Γεγονός το οποίο σηματοδοτεί την εκτύπωση του δένδρου μάχης του Μεγάλου Αλεξάνδρου. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

**Z**

**GA battle = <sid<sub>1</sub>, hid<sub>1</sub>>, <sid<sub>2</sub>, hid<sub>2</sub>>, <sid<sub>3</sub>, hid<sub>3</sub>>, ..., <sid<sub>n</sub>, hid<sub>n</sub>>**

**DONE**

όπως περιγράφηκε στο γεγονός **P**.

**W**

Γεγονός το οποίο σηματοδοτεί την εκτύπωση του πίνακα μάχης του Αριοβαρζάνη. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

**W**

**Index 0: <aid<sub>0,1</sub>>, <aid<sub>0,2</sub>>, ..., <aid<sub>0,N0</sub>>**

**Index 1: <aid<sub>1,1</sub>>, <aid<sub>1,2</sub>>, ..., <aid<sub>1,N1</sub>>**

**...**

**Index n: <aid<sub>n,1</sub>>, <aid<sub>n,2</sub>>, ..., <aid<sub>n,Nn</sub>>**

**DONE**

όπως περιγράφηκε στο γεγονός **A**.

## BONUS

Ως bonus, όποιος από τους φοιτητές επιθυμεί μπορεί να υλοποιήσει, **σε ξεχωριστό φάκελο κι ανεξάρτητα από την παραπάνω περιγραφή (δηλαδή επιπρόσθετα αυτής)**, το δένδρο μάχης του Μεγάλου Αλεξάνδρου ως ένα **AVL δένδρο** στο οποίο θα εκτελούνται **μόνο εισαγωγές**. Με άλλα λόγια, σε αυτό το παραδοτέο θα υλοποιούνται μόνο τα γεγονότα εκτύπωσης, τα γεγονότα **R** και **H** (ακριβώς όπως περιγράφηκαν παραπάνω) και το γεγονός **P**, με μόνη διαφορά ότι εδώ οι εισαγωγές θα γίνονται στο AVL δένδρο, αντί για το πλήρες δένδρο. Τα υπόλοιπα γεγονότα μπορούν να παραλειφθούν.

**Προσοχή**, όλα τα παραπάνω γεγονότα και οι δομές που περιγράφονται είναι **υποχρεωτικά**. Το bonus, σε περίπτωση που το παραδώσετε, θα πρέπει να βρίσκεται σε ξεχωριστό φάκελο και να εκτελείται ανεξάρτητα.

## Δομές Δεδομένων

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

```
struct Army {
    unsigned int id;
    struct Army *rc;
    struct Army *lc;
};

struct Horse {
    unsigned int id;
    unsigned int age;
};

struct HorsePQ {
    int size;
    struct Horse *HorsePQ;    // array that should be allocated dynamically
}

struct GA_Battle {
    unsigned int soldier_id;
    unsigned int horse_id
    struct GA_Battle *lc;
    struct GA_Battle *rc;
};

struct GA_Tree {
    int size;
    struct GA_Battle *GA_tree;
}

struct AR_Battle {
    unsigned int id;
    unsigned int type;
    struct AR_Battle *next;
};

struct AR_Battle **hash_table;

unsigned int max_soldiers_g;
unsigned int max_soldiers_id_g;
```

**Βαθμολογία Γεγονότων**

R	15
H	10
A	15
P	20
T	15
K	10
D	15
BONUS	15