



# Τεχνητή Νοημοσύνη

## Εργασία 1

Βασίλας Θωμάς

AM:3200012

Βλάσσης Γεώργιος

AM:3200020

Για την εκπόνηση της πρώτης εργασίας η ομάδα μας επέλεξε την εκπόνηση του ερωτήματος **α) Διάσχιση Ποταμού**, με τη χρήση του αλγορίθμου A\*. Η γλώσσα προγραμματισμού που χρησιμοποιήσαμε είναι η Java.

### Παρουσίαση του Προβλήματος

Στο παράδειγμά μας τα μέλη της οικογένειας(m) θα είναι 5, με χρόνους 1,3,6,8,12 και αναγνωριστικά (ID) 1,2,3,4,5 αντίστοιχα. Να σημειωθεί ωστόσο ότι το πρόγραμμα είναι κατασκευασμένο για να βρίσκει την βέλτιστη λύση για οποιοδήποτε πλήθος N ατόμων διθούν ως είσοδος, καθώς επίσης και τους χρόνους σε λεπτά που αντιστοιχούν στον καθένα αλλά και τον χρονικό περιορισμό. Σύμφωνα με τους κανόνες του παιχνιδιού, το επιθυμητό αποτέλεσμα του αλγορίθμου είναι για όλα μέλη να περάσουν στην άλλη πλευρά σε λιγότερο από 30 λεπτά. Το πρόβλημα ξεκινά με όλα τα μέλη όντας στη δεξιά πλευρά. Η κατάσταση αυτή ονομάζεται αρχική κατάσταση (initial state). Τα μέλη της οικογένειας μπορούν είτε μετακινηθείτε προς τα δεξιά ή προς τα αριστερά, ανάλογα με την τρέχουσα θέση τους. Κάθε τέτοια μετακίνηση αλλάζει και το state. Αφού στην αρχική κατάσταση όλα τα μέλη βρίσκονται στη δεξιά μεριά, στο πρώτο βήμα του αλγορίθμου θα

μετακινηθούν ένα ή δύο μέλη προς τα αριστερά. Οι καταστάσεις που παράγονται λόγω αυτού είναι και τα παιδιά(child) της αρχικής κατάστασης. Σε αυτό το σημείο διαπιστώνουμε ότι η καλύτερη λύση πρέπει να περιλαμβάνει 2 άτομα τη φορά κατά τη μετάβαση στα αριστερά και μόνο 1 άτομο στον κορμό κατά τη μετάβαση προς τα δεξιά, καθώς ο χρόνος κάθε μεταβίβασης ισούται με το ταχ των δύο μελών. Συνεπώς, θα ήταν σίγουρα μη αποδοτικό να μετακινείται στα αριστερά μόνο ένα άτομο τη φορά.

Η ευρετική που δημιουργήσαμε λειτουργεί θεωρώντας δεδομένο ότι το κόστος της κάθε κίνησης είναι μεγαλύτερο από το μηδέν, επομένως θα είναι συνεπής. Είναι γνωστό ότι ο αλγόριθμος A\* με κλειστό σύνολο και συνεπή και αποδεκτή ευρετική θα οδηγήσει σε μια βέλτιστη λύση. Ένας περιορισμός που καταργήσαμε είναι ότι ο αριθμός των μελών που μπορούν να διασχίσουν τον κορμό δεν υπερβαίνει τους δύο. Η ευρετική συνάρτηση υπολογίζει την αξία(value) κάθε κατάστασης(child) ως εξής:

- Εάν στην κατάσταση που θέλουμε να υπολογίσουμε η λάμπα βρίσκεται στην δεξιά πλευρά, τότε το h\_value θα ισούται με τον μέγιστο χρόνο εκ των μελών που βρίσκονται σε αυτή τη πλευρά.
- Εάν στην κατάσταση που θέλουμε να υπολογίσουμε η λάμπα βρίσκεται στην αριστερή πλευρά, τότε η αξία του είναι ξανά ίση με τον μέγιστο χρόνο εκ των μελών που βρίσκονται στα δεξιά συν τον ελάχιστο χρόνο των μελών που βρίσκονται στα αριστερά.

Τέλος, η συνάρτηση επιστρέφει έναν ακέραιο αριθμό που αντιπροσωπεύει το κόστος της μετάβασης από την τωρινή κατάσταση σε μια τερματική κατάσταση(terminal state). Με τη χρήση της συνάρτησης evaluate, προσθέτουμε την αξία αυτή με τον ήδη πεπερασμένο χρόνο για την επίτευξη της τωρινής κατάστασης. Έτσι επιλέγουμε τα child που έχουν το χαμηλότερη(στην περίπτωσή μας και βέλτιστη) αξία και μεταβαίνοντας στην ιδανική επόμενη κατάσταση συνθέτουμε την βέλτιστη δυνατή λύση.

## **Αρχιτεκτονική του Προγράμματος**

Το πρόγραμμά μας είναι γραμμένο σε Java και αποτελείται από τις εξής 4 κλάσεις:

- **FamilyMember:** Με την κλάση αυτή δημιουργούμε τα αντικείμενα FamilyMember που αντιπροσωπεύουν τα μέλη της οικογένειας του προβλήματος. Κάθε instance περιέχει ένα ID και τον χρόνο που χρειάζεται κάθε μέλος προκειμένου να διασχίσει το ποτάμι.
- **State:** Κάθε αντικείμενου τύπου State αντιπροσωπεύει την τρέχουσα κατάσταση στην οποία βρίσκονται τα μέλη της οικογένειας, συγκεκριμένα αν βρίσκονται αριστερά η δεξιά του ποταμού, καθώς και τη θέση της λάμπας. Εντός αυτής της κλάσης υλοποιείται και η ευρετική μας συνάρτηση που προαναφέραμε, καθώς και η συνάρτηση evaluate που χρησιμοποιείται για να βρεθεί η βέλτιστη κατάσταση στην οποία θα μεταβούμε. Επίσης περιέχεται και η compareTo, η οποία συγκρίνει τον βαθμό μεταξύ δύο καταστάσεων, αλλά και η show, η οποία απεικονίζει εικονικά στο terminal κάθε βήμα της βέλτιστης λύσης έως πέρας.
- **Main:** Εδώ τρέχουμε το πρόγραμμα. Αρχικά ζητείται από τον χρήστη η είσοδος των απαραίτητων δεδομένων, δηλαδή το πλήθος μελών της οικογένειας, τον χρόνο που αντιστοιχεί στον καθένα καθώς και τον χρονικό περιορισμό. Έπειτα υπολογίζεται η βέλτιστη λύση (εάν υπάρχει) και τυπώνεται το κάθε βήμα της βέλτιστης λύσης καθώς και επιπλέον πληροφορίες, όπως π.χ. τα δευτερόλεπτα που χρειάστηκε ο υπολογιστής για να καταλήξει στην λύση.
- **SpaceSearcher:** Εδώ υλοποιείται ο αλγόριθμος A\* για τον οποίον βασιστήκαμε κυρίως στον κώδικα του εργαστηρίου 1b.

## Βήμα προς βήμα η υλοποίηση του προβλήματος για τα δεδομένα που ζητήθηκαν

```
Input the number of family members:  
5  
Give the amount of minutes required to cross the river for the family member with id:1  
1  
Give the amount of minutes required to cross the river for the family member with id:2  
3  
Give the amount of minutes required to cross the river for the family member with id:3  
6  
Give the amount of minutes required to cross the river for the family member with id:4  
8  
Give the amount of minutes required to cross the river for the family member with id:5  
12  
Give the total time limit in minutes:  
30  
Finished in 7 steps.
```

### Αρχική κατάσταση

```
Initial State  
  
River  
Time : 0 mins
```

### 1<sup>o</sup> βήμα

```
Step 1: Moved from Right to Left  
  
Person 1 with time: 1 minutes  
Person 2 with time: 3 minutes  
  
(lampa) m1 m2 _____ River _____ m3 m4 m5  
Time : 3 mins
```

### 2<sup>o</sup> βήμα

```
Step 2: Moved from Left to Right  
  
Person 1 with time: 1 minutes  
  
m2 _____ River _____ (lampa) m3 m4 m5 m1  
Time : 4 mins
```

### 3<sup>ο</sup> βήμα

```
Step 3: Moved from Right to Left

Person 3 with time: 6 minutes
Person 1 with time: 1 minutes

(lampa) m2 m3 m1
----- River -----
m4 m5

Time : 10 mins
```

### 4<sup>ο</sup> βήμα

```
Step 4: Moved from Left to Right

Person 1 with time: 1 minutes

m2 m3
----- River -----
(lampa) m4 m5 m1

Time : 11 mins
```

### 5<sup>ο</sup> βήμα

```
Step 5: Moved from Right to Left

Person 4 with time: 8 minutes
Person 5 with time: 12 minutes

(lampa) m2 m3 m4 m5
----- River -----
m1

Time : 23 mins
```

### 6<sup>ο</sup> βήμα

```
Step 6: Moved from Left to Right

Person 2 with time: 3 minutes

m3 m4 m5
----- River -----
(lampa) m1 m2

Time : 26 mins
```

7<sup>ο</sup> βήμα

```
Step 7:Moved from Right to Left  
Person 1 with time: 1 minutes  
Person 2 with time: 3 minutes  
  
(lampa) m3 m4 m5 m1 m2  
----- River -----  
Time : 29 mins
```

Λεπτομέρειες λύσης

```
Minimum time for all members to cross the river: 29 minutes  
River crossed by all members within the time limit (30)  
Problem solved in : 0.007 sec
```

## Πειραματική τυχαία υποθολή δεδομένων

```
Input the number of family members:  
4  
Give the amount of minutes required to cross the river for the family member with id:1  
1  
Give the amount of minutes required to cross the river for the family member with id:2  
2  
Give the amount of minutes required to cross the river for the family member with id:3  
3  
Give the amount of minutes required to cross the river for the family member with id:4  
4  
Give the total time limit in minutes:  
15  
Finished in 5 steps.
```

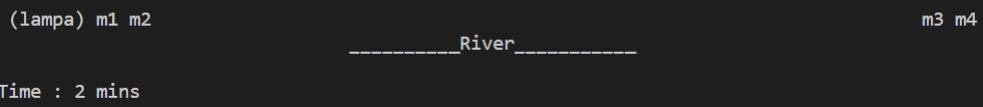
Αρχική κατάσταση

```
Initial State  
  
----- River -----  
(lampa) m1 m2 m3 m4  
Time : 0 mins
```

## 1<sup>o</sup> βήμα

Step 1: Moved from Right to Left

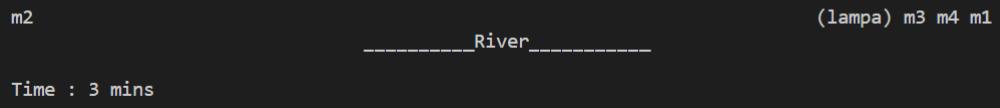
Person 1 with time: 1 minutes  
Person 2 with time: 2 minutes



## 2<sup>o</sup> βήμα

Step 2: Moved from Left to Right

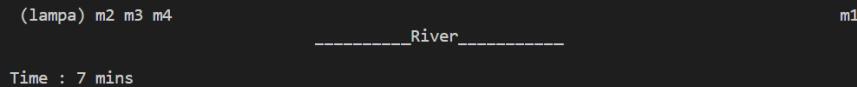
Person 1 with time: 1 minutes



## 3<sup>o</sup> βήμα

Step 3: Moved from Right to Left

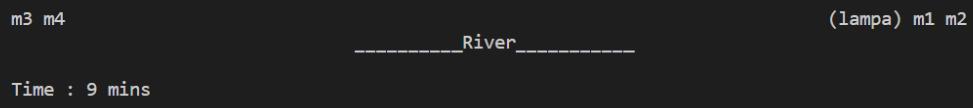
Person 3 with time: 3 minutes  
Person 4 with time: 4 minutes



## 4<sup>o</sup> βήμα

Step 4: Moved from Left to Right

Person 2 with time: 2 minutes



5<sup>ο</sup> βήμα

```
Step 5: Moved from Right to Left  
Person 1 with time: 1 minutes  
Person 2 with time: 2 minutes  
  
(lampa) m3 m4 m1 m2  
----- River -----  
Time : 11 mins
```

Λεπτομέρειες λύσης

```
Minimum time for all members to cross the river: 11 minutes  
River crossed by all members within the time limit (15)  
Problem solved in : 0.002 sec
```