

Πλάτων Καραγεώργης p3180068
Νικολέτα-Κλειώ Πατάτση p3180266
Ντενίς Ντιαμάντις p3150129

Τρόπος χρήσης

Το πρόβλημα με το οποίο ασχολήθηκε η ομάδα μας είναι η διάσχιση της γέφυρας γραμμένο σε κώδικα java. Το πρόγραμμα που δημιουργήσαμε βρίσκει την βέλτιστη σειρά με την οποία N άτομα με διαφορετικούς χρόνους διάσχισης πρέπει να περάσουν μια γέφυρα τη νύχτα. Για την ενέργεια αυτή είναι απαραίτητη η χρήση μίας λάμπα και η διάσχιση γίνεται με το πολύ δύο άτομα τη φορά με τον χρόνο του βραδύτερου.

Δυνατότητες

Οι είσοδοι του προγράμματος είναι ο αριθμός των ατόμων, τα ονόματα τους και ο χρόνος π.χ. σε λεπτά που χρειάζεται ο καθένας και ως έξοδο εκτυπώνει στην οθόνη τη σειρά με την οποία πρέπει να περάσουν και τον ελάχιστο χρόνο που απαιτείται.

Αρχιτεκτονική και μέθοδος Τεχνητής Νοημοσύνης

Για την αναπαράσταση των στιγμιοτύπων των καταστάσεων του προβλήματος χρησιμοποιήσαμε δύο λίστες τύπου ArrayList με τους χρόνους των ατόμων και μια Boolean lamp που παριστάνουν τις όχθες και την τοποθεσία της λάμπα. Ξεκινάμε από την αρχική κατάσταση όπου όλοι βρίσκονται στη δεξιά όχθη. Με την χρήση της μεθόδου getChildren() δημιουργούνται τα παιδιά της τωρινής κατάστασης ανάλογα με την τοποθεσία της λάμπα. Ως ευρετική θεωρήσαμε πως από δεξιά στα αριστερά περνούν πάντα δύο άτομα ενώ απο αριστερά στα δεξιά μόνο ο ταχύτερος. Εάν η λάμπα βρίσκεται δεξιά(έχει την τιμή true) τότε αποθηκεύουμε προσωρινά την τωρινή κατάσταση και με δύο for loops παράγουμε όλους τους δυνατούς συνδυασμούς των ατόμων μοναδικά(για παράδειγμα παράγεται το παιδί που περνά με συνδιασμό χρόνων 1-3 όχι όμως και το 3-1 διότι οδηγούν στην ίδια κατάσταση). Για κάθε παιδί υπολογίζεται ο χρόνος του βραδύτερου συν τον χρόνο του πατέρα του από την ρίζα. Στην περίπτωση που η λάμπα είναι αριστερά(έχει την τιμή false) επιλέουμε τον ταχύτερο να περάσει και δημιουργείται ένα παιδί με τον χρόνο του ταχύτερου συν τον χρόνο του πατέρα του από την ρίζα.

Με την χρήση του αλγορίθμου A* καλείται επαναληπτικά η μέθοδος getChildren() και αφού γίνει η σύγκριση των καταστάσεων ανάλογα με τον χρόνο επιλέγεται να διερευνηθεί η κατάσταση με τον μικρότερο χρόνο μέχρι φτάσουμε στην τελική κατάσταση όπου η δεξιά όχθη θα είναι άδεια.

Τέλος εκτυπώνονται στην οθόνη οι καταστάσεις από την αρχική έως την τελική και ο συνολικός χρόνος που χρειάστηκε.

Παραδείγματα

Για $N=5$ και χρόνους 1,3,6,8,12 το αποτέλεσμα έχει συνολικό χρόνο 29 και εκτελείται σε 0.034 sec.

Για $N=7$ και χρόνους 1,3,6,8,12,15,18 το αποτέλεσμα έχει συνολικό χρόνο 54 και εκτελείται σε 5.002 sec.

Για $N=8$ και χρόνους 1,3,6,8,12,15,18,21 το αποτέλεσμα έχει συνολικό χρόνο 68 και εκτελείται σε 1304.029 sec \sim 21 min.