

Missioners and Cannibals

Τρόπος χρήσης:

Το πρόγραμμα δέχεται τις τιμές N (πλήθος κανιβάλων και ιεραποστολών), M μέγιστος αριθμός θέσεων στη βάρκα και K τον μέγιστο αριθμό μετακινήσεων της βάρκας ως είσοδο στον κατασκευαστή State (int N, int M, int K).

Αρχιτεκτονική:

Το πρόγραμμα αποτελείται από 3 κλάσεις από τη Main στην οποία γίνεται η ανάθεση τιμών η κλήση του αλγορίθμου επίλυσης και η χρονομέτρηση του προγράμματος. Έπειτα έχουμε τη State στην οποία γίνεται η δημιουργία των παιδιών του κόμβου που ορίσαμε στη Main αλλά και των παιδιών των παιδιών του κλπ. Ο έλεγχος αν είναι έγκυρο το παιδί (δεν παραβιάζεται κάποιος από τους περιορισμούς). Όπως και η ανάθεση identifier σε κάθε παιδί. Τέλος, η SpaceSearcher κάνει την αναζήτηση στο δέντρο για τη βέλτιστη λύση με χρήση αλγορίθμου A*

Μέθοδοι τεχνίτης νοημοσύνης που χρησιμοποιήθηκαν

Η αναζήτηση της βέλτιστης λύσης έγινε με τον αλγόριθμο αναζήτησης A* ($f(n)=h(n)+g(n)$). Η εκτέλεση έγινε με $h(n)$ = BestFS όπου διάλεγε την καλύτερη επομένη κατάσταση στον κόμβο n με τη βοήθεια της evaluate η οποία υπολογίζει πια είναι η «αξία» ενός κόμβου σύμφωνα με τις κινήσεις που θα έπρεπε να κάνει η βάρκα αν δεν υπήρχαν καθόλου περιορισμοί. Ως $g(n)$ έχουμε το κόστος από τη ρίζα ως το n

Πειραματικά αποτελέσματα

Οι τιμές που δοκιμάστηκαν θα εκφραστούν παρακάτω στη μορφή:

(αριθμός κανιβάλων *, Θέσεις βάρκας, βήματα για εύρεση λύσης, χρόνος)

Τα αποτελέσματα:

(2, 2, 5, 0.001s)

(3, 2, 11, 0.001s)

(8, 4, 13, 0.001s)

(16, 4, 29, 0.003s)

(32, 4, 61, 0.016s)

(64, 4, 125, 0.464s)

(100, 7, 80, 120s) Max steps reached

(100, 10, 49, 51.593s)

* έχουμε υποθέσει ότι έχουμε ίσο αριθμό ιεραποστόλων και κανιβάλων

Σημείωση: Οι παραπάνω τιμές υπολογίστηκαν εντός το IDE IntelliJ σε laptop με χαρακτηριστικά (R5-3500u, 16 GB RAM)