

ΕΦΑΡΜΟΣΜΕΝΗ ΣΥΝΔΥΑΣΤΙΚΗ ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΙΑ

ΟΝΟΜΑ: ΠΑΝΑΓΙΩΤΗΣ

ΕΠΩΝΥΜΟ: ΚΕΛΕΣΙΔΗΣ

ΑΜ: Π23236

ΠΕΡΙΓΡΑΦΗ ΥΛΟΠΟΙΗΣΗΣ:

Ο στόχος του αλγορίθμου είναι η εύρεση της ελάχιστης διδρομής που χρειάζεται να διανύσει ένας πλανόδιος πωλητής για την επίσκεψη 10 πόλεων. Για την υλοποίηση χρησιμοποιείται αλγόριθμος του τύπου knight-tour. Αρχικά οι αποστάσεις μεταξύ των πόλεων αποθηκεύονται σε έναν δυσδιάστατο πίνακα. Στη συνέχεια εισάγεται η πρώτη πόλη σε λίστα και επιλέγεται η επόμενη πόλη που θα εισαχθεί στη τελική ελάχιστη απόσταση. Η επιλογή γίνεται με βάση την μικρότερη απόσταση που απέχει η πόλη που ελέγχουμε με την αμέσως επόμενη. Οι ελάχιστες αποστάσεις προστίθενται μεταξύ τους και τυπώνονται ως η ελάχιστη διαδρομή.

ΨΕΥΔΟΚΩΔΙΚΑΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ:

1) void createDistanceMatrix()

Κατασκευή τυχαίου δυσδιάστατου πίνακα. Για κάθε σειρά και στήλη του πίνακα allDistances[][] τοποθετείται τυχαίος αριθμός. Στην κύρια διαγώνιο τοποθετείται το 0. Λόγω της φύσης του προβλήματος ο πίνακας είναι συμμετρικός, το οποίο επιτυγχάνεται εδώ : allDistances[i][j] = r; allDistances[j][i] = r;

2) class City

Κατασκευή κλάσης πόλη. Αρχικοποίηση λίστας που περιέχει τις αποστάσεις της πόλης από όλες τις υπόλοιπες. Επιπλέον χρησιμοποιείται μια μεταβλητή x τύπου bool που συγκρατεί αν έχει επισκεφθεί η πόλη (στον constructor η x αρχικοποιείται ως false αφού καμία πόλη δεν έχει επισκευθεί εξ αρχής). Επιπλέον η συνάρτηση string printCity() τυπώνει τον αριθμό κάθε πόλης (πχ city 1, city 2)

3) void createCities()

Αρχικοποίηση πινάκων ανα πόλη. Με τη χρήση nested for loops γίνεται προσπέλαση του πίνακα με τις αποστάσεις και κάθε στοιχείο της κάθε γραμμής του αποθηκεύεται στον πίνακα της αντίστοιχης πόλης. Για παράδειγμα αν η πρώτη γραμμή του πίνακα αποστάσεων είναι {0, 1, 2, 3} τότε η λίστα της πόλης city 1 θα λάβει τις τιμές 0, 1, 2, 3

4) int findMin(vector<int> v)

Υπολογισμός ελάχιστης απόστασης. Προσπέλαση της δοθείσας λίστας και αναζήτηση της επόμενης πόλης η οποία πρέπει να τηρεί τα εξής 2 κριτήρια:

1. Η απόστασή της να είναι η ελάχιστη(φυσικά εξαιρείται το 0)
2. Να μην την έχουμε ξαναεπισκευτεί

Αν πληρούνται όλα τα παραπάνω κριτήρια η αντίστοιχη πόλη εισάγεται στη διαδρομή, η μεταβλητή x λαμβάνει την τιμή true και επιστρέφεται η απόσταση η οποία βρέθηκε να είναι η ελάχιστη.

5) void findBestRoute()

Εύρεση βέλτιστης διαδρομής. Τοποθέτηση πρώτης πόλης στην τελική διαδρομή και η x γίνεται true. Στη συνέχεια υπολογίζεται η ελάχιστη απόσταση απο κάθε πόλη στην επόμενη και προστίθεται στην τελική ελάχιστη απόσταση. Στο τέλος προστίθεται ξανά η αρχική πόλη και η αντίστοιχη απόσταση απο την προηγούμενη πόλη.

6) int main()

Εκτέλεση όλων των παραπάνω συναρτήσεων.

SCREENSHOT ΑΠΟ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ:

```
PS C:\Users\pkele\Desktop\ΠΑΝΕΠΙΣΤΗΜΙΟ\4TH SEMESTER\ΕΦΑΡΜΟΣΜΕΝΗ ΣΥΝΔΥΑΣΤΙΚΗ\Erg1> .\ERGASIA1.exe
0      45      61      26      3       79      64      71      56      11
45     0       85      31      94      92      92      100     42      87
61     85      0       55      30      7       23      26      42      85
26     31      55      0       51      85      20      46      29      63
3      94      30      51      0       16      29      9       60      36
79     92      7       85      16      0       58      10      28      27
64     92      23      20      29      58      0       27      6       41
71     100     26      46      9       10      27      0       65      52
56     42      42      29      60      28      6       65      0       44
11     87      85      63      36      27      41      52      44      0
city1 -> city2 -> city4 -> city5 -> city3 -> city9 -> city6 -> city7 -> city8 -> city10 -> city1
The minimum distance required to travel is: 321
```

