

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

1^η εργασία

2017-2018

Στόχος της εργασίας αυτής είναι η υλοποίηση ενός αλγορίθμου που θα βρίσκει ημερήσια δρομολόγια για τουρίστες που επισκέπτονται ένα προορισμό. Ο αλγόριθμος που θα υλοποιηθεί είναι μία απλοποιημένη εκδοχή αλγορίθμων που έχουν προταθεί στη βιβλιογραφία.

Περιγραφή προβλήματος

Έστω ένας τουρίστας ο οποίος επισκέπτεται ένα προορισμό για k ημέρες. Ο προορισμός αυτός έχει λ αξιοθέατα $\{\alpha_1, \alpha_2, \dots, \alpha_\lambda\}$ και ο χρήστης έχει βαθμολογήσει τα αξιοθέατα με κάποια βαθμολογία. Συγκεκριμένα, το αξιοθέατο α_i έχει βαθμολογηθεί με βαθμολογία β_i . Επίσης κάθε αξιοθέατο α_i έχει ώρες λειτουργίας $αν_i$ - $κλι$ ενώ από ιστορικά στοιχεία έχει καθορισθεί ότι ο μέσος χρόνος επίσκεψης είναι $χει$. Για να είναι εφικτή μία επίσκεψη σε ένα αξιοθέατο θα πρέπει η άφιξη να πραγματοποιηθεί πριν την ώρα που κλείνει το αξιοθέατο και η ολοκλήρωση της επίσκεψης (λαμβάνοντας υπόψη και το μέσο χρόνο επίσκεψης) πρέπει επίσης να είναι πριν την ώρα αυτή. Αν ο τουρίστας φθάσει στο αξιοθέατο πριν την ώρα που ανοίγει, περιμένει στην είσοδο του αξιοθέατου μέχρι αυτό να ανοίξει.

Επιπλέον, υπάρχει η πληροφορία των συντεταγμένων για κάθε αξιοθέατο και η απόσταση $απ_{ij}$ μεταξύ δύο αξιοθεάτων α_i και α_j δίνεται από την ευκλείδεια απόσταση των αντίστοιχων συντεταγμένων. Για ευκολία, θεωρήστε ότι η $απ_{ij}$ είναι ο χρόνος που απαιτείται για να μεταβεί κάποιος από το α_i στο α_j .

Για κάθε μία από τις k ημέρες διαμονής, ο τουρίστας μπορεί να διαθέσει κάποιες ώρες για να δει τα αξιοθέατα του τουριστικού προορισμού: την i -οστή ημέρα διαμονής, ο τουρίστας μπορεί να δει αξιοθέατα μεταξύ των ωρών $αρχ_i$ και $τελ_i$. Κάθε περιήγηση αρχίζει και ολοκληρώνεται στο ξενοδοχείο του τουρίστα των οποίων οι συντεταγμένες πάλι δίνονται.

Δεδομένων των παραπάνω περιορισμών, στόχος είναι η εύρεση k δρομολογίων ένα για κάθε ημέρα διαμονής τα οποία θα διέρχονται από τα αξιοθέατα που έχουν την μεγαλύτερη βαθμολογία. Πιο αναλυτικά, το άθροισμα των βαθμολογιών όλων των αξιοθεάτων που περιλαμβάνονται στα k δρομολόγια της λύσης πρέπει να είναι το μέγιστο δυνατό.

Προτεινόμενος αλγόριθμος

Αρχικά ο αλγόριθμος ξεκινά με k άδεια δρομολόγια που αρχίζουν και ολοκληρώνονται στο ξενοδοχείο του τουρίστα. Ο αλγόριθμος εκτελεί στη συνέχεια $\#επαν$ επαναλήψεις, όπου η παράμετρος $\#επαν$ δίνεται ως είσοδο στον αλγόριθμο.

Έστω ότι ο αλγόριθμος εκτελεί μία από αυτές τις επαναλήψεις (έστω την j -οστή επανάληψη) και $\Delta_1, \Delta_2, \dots, \Delta_k$ τα δρομολόγια που έχει προσδιορίσει ο αλγόριθμος ακριβώς πριν την επανάληψη αυτή. Στην τρέχουσα επανάληψη, αλγόριθμος θα προσπαθήσει να επεκτείνει τα δρομολόγια αυτά με όσο το δυνατόν περισσότερα νέα αξιοθέατα εφόσον 1) οι νέες προσθήκες δεν μετατρέπουν σε ανέφικτες τις

επισκέψεις σε αξιοθέατα που είναι ήδη στα δρομολόγια και 2) ο συνολικός χρόνος περιήγησης για κάθε δρομολόγιο δεν υπερβαίνει το διαθέσιμο χρόνο που έχει ο τουρίστας την αντίστοιχη ημέρα.

Το βασικό κριτήριο που καθορίζει την επιλογή ενός αξιοθέατου α_i ως το επόμενο που θα εισαχθεί σε κάποιο δρομολόγιο είναι το όφελος που προκύπτει από την εισαγωγή του στο δρομολόγιο αυτό. Συγκεκριμένα, ας επιχειρήσουμε να τοποθετήσουμε το α_i μεταξύ δύο διαδοχικών αξιοθέατων α_q και α_r ενός δρομολογίου. Το όφελος που δημιουργείται από αυτή την εισαγωγή, εφόσον αυτή είναι εφικτή και δεν δημιουργεί ακυρώσεις επισκέψεων σε άλλα ήδη επιλεγμένα αξιοθέατα, ορίζεται ως

$$O\varphi_i = \frac{\beta_i^2}{\alpha\pi_{qi} + \chi\epsilon_i + \alpha\pi_{ir} - \alpha\pi_{qr}}$$

όπου ο παρονομαστής δείχνει την αύξηση του χρόνου περιήγησης που προκαλείται στο δρομολόγιο μετά την επιχειρούμενη εισαγωγή.

Έτσι, για κάθε αξιοθέατο α_i εξετάζεται κάθε πιθανή θέση εισαγωγής στα δρομολόγια $\Delta_1, \Delta_2, \dots, \Delta_k$ και καθορίζεται το δρομολόγιο και η θέση όπου προκύπτει το μέγιστο όφελος. Στη συνέχεια, το αξιοθέατο με το μέγιστο όφελος σε σχέση με τα υπόλοιπα αξιοθέατα, εισάγεται στη θέση του μέγιστου οφέλους.

Η παραπάνω διαδικασία επαναλαμβάνεται για τα εναπομείναντα αξιοθέατα μέχρι καμία άλλη εισαγωγή να είναι εφικτή.

Προκειμένου, ο αλγόριθμος να μην εγκλωβιστεί σε μία τοπικά βέλτιστη λύση, ως τελευταίο βήμα της j -οστής επανάληψης, εκτελείται μία διαγραφή αξιοθέατων από τα διαμορφωμένα μέχρι εκείνη τη στιγμή δρομολόγια. Συγκεκριμένα, για κάθε δρομολόγιο Δ_i επιλέγεται τυχαία ένα αξιοθέατο του και με αρχή αυτό διαγράφονται π^* (πλήθος αξιοθέατων του Δ_i) συνεχόμενα αξιοθέατα με κατεύθυνση προς το ξενοδοχείο και όπου π είναι μία παράμετρος εισόδου με τιμές στο $(0,1)$. Αν τα αξιοθέατα που πρέπει να διαγραφούν είναι περισσότερα από αυτά που μεσολαβούν μέχρι το ξενοδοχείο, η διαγραφή συνεχίζει και στα αξιοθέατα τα οποία είναι πέραν του ξενοδοχείου.

Μετά την διαγραφή αυτή, η j -οστή επανάληψη έχει ολοκληρωθεί και ο αλγόριθμος προχωράει στην $(j+1)$ -επανάληψη.

Ο αλγόριθμος αυτός βρίσκει μία προσεγγιστική λύση στο πρόβλημα που στη πράξη είναι κοντά στη βέλτιστη. Εσείς θα πρέπει να υλοποιήσετε τον αλγόριθμο σε C++.

Είσοδος- Έξοδος Αλγορίθμου

Για να μπορέσετε να ελέγξετε τον αλγόριθμο που θα υλοποιήσετε στο αρχείο `test_instances.zip`, θα βρείτε αρχεία που περιέχουν διάφορα αντιπροσωπευτικά σενάρια και μπορείτε να τα χρησιμοποιήσετε ως είσοδο στον αλγόριθμο σας. Επίσης στο ίδιο αρχείο θα βρείτε και πληροφορίες για τη γραμμογράφηση (format) αυτών των αρχείων. Ως έξοδο, το πρόγραμμα σας θα τυπώνει τα δρομολόγια που έχει καθορίσει ο αλγόριθμος.

Παραδοτέα

Θα πρέπει να παραδοθεί ο πηγαίος κώδικας εκτυπωμένος καθώς και σε CD μαζί με τον εκτελέσιμο κώδικα. Ιδιαίτερη βαρύτητα θα πρέπει να δοθεί στη σωστή τεκμηρίωση του προγράμματός σας. Θα πρέπει λοιπόν ο κώδικας σας να συνοδεύεται από ξεχωριστό κείμενο που θα παρέχει λεπτομερή περιγραφή των τεχνικών σας. Επίσης, εντός του πηγαίου κώδικα θα πρέπει να υπάρχουν «πυκνά» σχόλια διατυπωμένα στα ελληνικά.

Η εργασία μπορεί να εκπονηθεί από ομάδα μέχρι δύο ατόμων αυστηρώς.

Προθεσμία Παράδοσης: Δευτέρα 30 Απριλίου 2018, θυρίδα διδάσκοντος.