

Group Project LSO 2021-22

Teams: 2-4 Members

Grade: 30% of the Final Grade

Deadline: January 2022 (Exact date to be announced before Christmas holidays)

Submission: Each group must have a leader responsible for submitting the deliverables. Submitted deliverables must include all group member names.

Περιγραφή Προβλήματος

Θεωρήστε μία κεντρική αποθήκη (Κόμβος με id: 0) και ένα σύνολο $n = 300$ πελατών (Κόμβοι με id: 1,...,n=300).

Όλοι οι κόμβοι βρίσκονται σε ένα τετράγωνο πλευράς 100. Θεωρήστε πως ο χρόνος μετάβασης από κόμβο σε κόμβο είναι ίσος με την Ευκλείδεια απόσταση μεταξύ των δύο κόμβων.

Κάθε πελάτης i έχει έναν απαιτούμενο χρόνο εξυπηρέτησης st_i και ένα κέρδος p_i .

Ένας στόλος $k = 5$ φορτηγών αυτοκινήτων βρίσκεται στην κεντρική αποθήκη.

Τα οχήματα ξεκινούν από την αποθήκη, εξυπηρετούν πελάτες και κατόπιν επιστρέφουν πίσω στην κεντρική αποθήκη.

Κάθε όχημα εκτελεί μία διαδρομή.

Κάθε πελάτης μπορεί να καλυφθεί (δεν είναι αναγκαίο πως θα καλυφθεί) από μία επίσκεψη ενός και μόνο οχήματος. Στην περίπτωση αυτή, ο πελάτης αποδίδει το κέρδος του.

Ο συνολικός χρόνος κάθε διαδρομής (χρόνος μεταβάσεων και χρόνος εξυπηρέτησης πελατών) δε μπορεί να υπερβαίνει ένα χρονικό όριο $T = 150$.

Σκοπός του προβλήματος είναι ο σχεδιασμός k διαδρομών οι οποίες θα μεγιστοποιούν το συνολικό κέρδος. Προφανώς, λόγω των περιορισμών του μέγιστου χρονικού ορίου δεν είναι απαραίτητο πως θα καλυφθούν όλοι οι πελάτες. Αντίθετα, πρέπει να επιλεγούν και να δρομολογηθούν οι επιλεγμένοι πελάτες.

Παραδοτέα Εργασίας

A. Γράψτε σε python τις κλάσεις, πίνακες και γενικότερα την απαιτούμενη υποδομή για αναπαρασταθεί το πρόβλημα (3).

Οι κόμβοι αποθήκη και πελάτες να δημιουργούνται και να φυλάσσονται όπως φαίνεται στον παρακάτω κώδικα:

```
all_nodes = []
d = Node(0, 50, 50, 0, 0)
all_nodes.append(d)
birthday = 5112001
random.seed(birthday)
for i in range(0, 300):
    xx = random.randint(0, 100)
    yy = random.randint(0, 100)
    service_time = random.randint(5, 10)
    profit = random.randint(5, 20)
    cust = Node(i+1, xx, yy, service_time, profit)
    all_nodes.append(cust)
```

όπου birthday η ημέρα γενεθλίων του αρχηγού της ομάδας.

Β. Φτιάξτε έναν κατασκευαστικό αλγόριθμο ο οποίος θα σχηματίζει μία ολοκληρωμένη λύση (3).

Γ. Γράψτε σε Python τέσσερις τελεστές τοπικής έρευνας (3)

1. **Relocate**: Επανατοποθέτηση κάθε καλυπτόμενου πελάτη σε οποιοδήποτε διαφορετικό σημείο της λύσης.
2. **Swap**: Αντιμετάθεση των θέσεων εξυπηρέτησης οποιουδήποτε ζεύγους καλυπτόμενων πελατών.
3. **Insertion**: Εισαγωγή ενός οποιουδήποτε μη καλυπτόμενου πελάτη σε οποιοδήποτε σημείο της λύσης.
4. **Profitable Swap**: Αντικατάσταση οποιουδήποτε καλυπτόμενου πελάτη με οποιονδήποτε μη καλυπτόμενο πελάτη.

Δ. Γράψτε σε Python μία VND μέθοδο η οποία θα χρησιμοποιεί τους τελεστές του ερωτήματος (Γ) και θα βελτιώνει τη λύση του ερωτήματος (Β). Ποια είναι η τελική λύση. Ποιο το κόστος της και μετά από πόσες επαναλήψεις του αλγορίθμου δημιουργήθηκε (1);