

Τίτλος:

2. Εργασία με χρήση ΑΡΑΙΩΝ ΜΗΤΡΩΝ (SPARSITY TECHNOLOGY) (2020-21)

Περιγραφή:

Δημιουργήστε 10.000 τυχαία ταξίδια (αλυσίδες συνδέσεων, legs) ανάμεσα στους κόμβους ($V=800$) ενός γράφου.

Μπορείτε να σκεφτείτε το πρόβλημα και σαν ένα σύνολο αεροδρομίων για ένα χρήστη ιδιωτικού αεροπλάνου!

Το βάρος (πχ χρόνος πτήσης) ανάμεσα σε κάθε ζευγάρι κόμβων είναι ένας τυχαίος ακέραιος αριθμός από 50 έως 100 και δεν είναι συμμετρικό!

Δημιουργήστε με τυχαίο τρόπο και αποθηκεύστε όλα τα βάρη ανάμεσα στους κόμβους.

Ένα ταξίδι πρέπει να έχει συνολικό βάρος (άθροισμα βαρών πτήσεων) μεγαλύτερο από 200.

Για να φτιάξουμε ένα ταξίδι αρχίζουμε με έναν τυχαίο κόμβο i ($1-V$) και έναν τυχαίο κόμβο j ($1-V$) και $val(i,j)$ (από τον πίνακα που ήδη δημιουργήσαμε) για την σύνδεση (i,j) .

Συνεχίζουμε με την τυχαία τιμή k ($1-V$) & $val(j,k)$ για την επόμενη σύνδεση (πτήση) (j,k) , έως ότου έχουμε περάσει το κάτω όριο του βάρους ενός ταξιδιού.

Η αρχικοποίηση της γεννήτριας τυχαίων αριθμών να γίνεται με χρήση του αριθμού μητρώου σας.

Δεν πρέπει σε ένα ταξίδι να περάσουμε από τον ίδιο κόμβο δυο φορές και δεν είναι υποχρεωτικό να γυρίσουμε στο σημείο (αεροδρόμιο) που ξεκινήσαμε!

Η μήτρα $A(m \times n)$ που ισοδυναμεί με την ως άνω περιγραφή είναι αραιά και το $m = V$ είναι οι κόμβοι (αεροδρόμια) και $n = 10,000$ είναι τα ταξίδια που θέλουμε να δημιουργήσουμε με την μεθοδολογία που εξηγήσαμε.

Δημιουργούμε τις στήλες της μήτρας A που ουσιαστικά αντιστοιχούν σε ταξίδια από τη στήλη 1 έως τη στήλη 10,000 με τη σειρά!

Αυτή η διαδικασία συχνά αναφέρεται ως "generator".

Ο πλήρης πίνακας $A(m \times n)$ ΔΕΝ πρέπει να δημιουργηθεί!

α) Αποθηκεύστε την αραιά μήτρα $A(m \times n)$ σύμφωνα με τον τρόπο 1 που είπαμε στο μάθημα,

δηλαδή με χρήση του πίνακα $a1_iloc$ (μεγέθους 10,000) με την θέση $a1_iloc(q)$ να μας δίνει το index πρόσβασης στους κόμβους του ταξιδιού q στον πίνακα $a1_irow$. Ο πίνακας $a1_irow$ (μεγέθους περίπου $10,000 \times 5$) περιέχει όλους τους κόμβους που επισκέπτονται τα ταξίδια.

Ο πίνακας $a1_jval$ (μεγέθους 10,000) περιέχει το βάρος του κάθε ταξιδιού που πρέπει και να αλλάζει όταν προσθέτουμε ή αφαιρούμε συνδέσεις (legs) και συνεπώς προορισμούς σε ένα ταξίδι.

β) Αποθηκεύστε την αραιά μήτρα $A(m \times n)$ σύμφωνα με τον τρόπο 2 που είπαμε στο μάθημα,

δηλαδή χρήση των πινάκων $a2_iloc$, $a2_jval$, $a2_irow$ και $a2_next$ που είναι πιο ευέλικτος στην πρόσθεση και αφαίρεση εκ των υστέρων συνδέσεων (legs) στα ταξίδια μας.

Στη συνέχεια και για τους δύο τρόπους:

1. Υλοποιήστε αλγόριθμο που προσθέτει στο ταξίδι 1000 έναν τυχαίο προορισμό, που δεν υπάρχει ήδη στο ταξίδι αυτό.

Να υπάρχει όρισμα που να ορίζει αν ο νέος προορισμός πρέπει να εισαχθεί στην αρχή ή στο τέλος του ταξιδιού και να αλλάζει το βάρος του ταξιδιού.

2. Υλοποιήστε αλγόριθμο που υπολογίζει το γινόμενο του ανάστροφου πίνακα AT ($n \times m$) επί τον πίνακα A ($m \times n$).

Δηλαδή να υπολογισθεί το $B = AT * A$ διαστάσεων ($n \times n$) και να αποθηκευτεί σε δομή αραιάς μήτρας με χρήση των πινάκων b_iloc (μεγέθους n), b_jcol και b_val .

Θεωρούμε ότι η κάθε στήλη του πίνακα A , που όπως είπαμε αντιστοιχεί σε ένα ταξίδι,

έχει την τιμή 1 για κάθε κόμβο (αεροδρόμιο) που επισκέπτεται και έχει αποθηκευτεί στον πίνακα a_row .

Όταν πολλαπλασιάζουμε τις k & l στήλες του A (που αντιστοιχούν στα ταξίδια k & l) η τιμή $b_val(k,l)$ δείχνει πόσους κοινούς προορισμούς περιέχουν!

Ποια είναι η πολυπλοκότητα $O(?)$ του υπολογισμού της μήτρας B ?

Ποιο είναι το μέγεθος των πινάκων b_col & b_val ?

3. Προσδιορίστε τον κόμβο με τη μικρότερη επισκεψιμότητα.

Εξηγήστε τον αλγοριθμικό τρόπο που χρησιμοποιήσατε για την εύρεση αυτού του κόμβου.

4. Δημιουργείστε διπλάσια (20,000) ταξίδια και σχολιάστε πως μεταβάλλεται ο χρόνος στα ερωτήματα 2 και 3.

Παραδίδετε συμπιεσμένο αρχείο με το username σας (π.χ up123456.zip) που περιέχει τα αρχεία του κώδικα (σε source file) και την αναφορά σε pdf ή txt file (παρακαλώ όχι doc), που περιγράφει τη δομή και τους αλγόριθμους που χρησιμοποιήθηκαν για την αποθήκευση των δεδομένων και για την απάντηση των ερωτήσεων.

Ο κώδικας που θα παραδώσετε πρέπει να περιέχει τα απαραίτητα για την κατανόησή του σχόλια και πριν από κάθε συνάρτηση (ή μέθοδο) να υπάρχει μια σύντομη περιγραφή της λειτουργίας της.