



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Αλγόριθμοι και Πολυπλοκότητα

Διδάσκοντες: Σ. Ζάχος, Δ. Φωτάκης

3η Σειρά Προγραμματιστικών Ασκήσεων - Ημ/νία Παράδοσης 15/1/2015

Άσκηση 1: Μετακινήσεις

Στη μακρινή Χώρα των Αλγορίθμων υπάρχουν N πόλεις που συνδέονται μεταξύ τους με M λεωφόρους διπλής κατεύθυνσης. Κάθε λεωφόρος συνδέει δύο πόλεις. Μάλιστα η μετακίνηση στις λεωφόρους είναι τόσο γρήγορη που μπορούμε να θεωρήσουμε ότι γίνεται στιγμιαία! Έτσι είναι συνηθισμένο οι κάτοικοι να διαμένουν σε διαφορετική πόλη από αυτή στην οποία εργάζονται.

Οι λεωφόροι, δυστυχώς, είναι παλιές και χρειάζονται καθημερινή συντήρηση, με αποτέλεσμα να κλείνουν από μία χρονική στιγμή και μετά. Ξέρουμε ότι η i -οστή λεωφόρος κλείνει για συντήρηση τη χρονική στιγμή $d_i > 0$, και συνεπώς μπορεί να χρησιμοποιηθεί για μετακινήσεις μόνο κατά τις χρονικές στιγμές $t \in [0, d_i]$. Βέβαια οι λεωφόροι έχουν σχεδιαστεί ώστε τη χρονική στιγμή 0, όλοι να μπορούν να μεταβούν στην εργασία τους.

Οι εκλογές πλησιάζουν, και ο Πρόεδρος της χώρας σας έχει αναθέσει να βελτιώσετε τις συγκοινωνίες (και τη ζωή των ψηφοφόρων). Το χρήματα που έχετε στη διάθεσή σας επαρκούν για K οδικά έργα. Κάθε οδικό έργο περιλαμβάνει είτε την πλήρη ανακατασκευή μιας υπάρχουσας λεωφόρου είτε τη δημιουργία μιας νέας. Μετά από αυτό, η αντίστοιχη λεωφόρος δεν θα χρειάζεται πλέον συντήρηση, και θα μπορεί να χρησιμοποιηθεί για μετακινήσεις οποιαδήποτε χρονική στιγμή. Στόχος αυτών των έργων είναι να μεγιστοποιήσουμε την τελευταία χρονική στιγμή T^* όπου εξασφαλίζεται ότι όλοι μπορούν να μεταβούν στην εργασία τους (ή ισοδύναμα, εξασφαλίζεται ότι η συνδεσιμότητα της χώρας είναι αντίστοιχη με τη συνδεσιμότητα κατά τη χρονική στιγμή 0). Αυτό είναι σημαντικό γιατί οι κάτοικοι δεν θα χρειάζεται να ξεκινήσουν για την εργασία τους πριν τη χρονική στιγμή T^* , και έτσι θα έχουν περισσότερο χρόνο για να ξεκουραστούν. Πρέπει λοιπόν να υπολογίσετε τη μέγιστη χρονική στιγμή T^* που μπορεί να επιτευχθεί με K οδικά έργα. **Σημείωση:** Προσέξτε ότι δεν γνωρίζουμε ποιος δουλεύει πού. Γνωρίζουμε μόνο ότι τη χρονική στιγμή 0, ο καθένας μπορεί να μεταβεί από την κατοικία στην εργασία του. Αυτό δεν σημαίνει απαραίτητα ότι τη χρονική στιγμή 0 μπορούμε να μεταβούμε από κάθε πόλη σε οποιαδήποτε άλλη!

Δεδομένα Εισόδου: Το πρόγραμμά σας θα διαβάζει από το standard input τρεις θετικούς ακέραιους N , M , και K , που αντιστοιχούν στο πλήθος των πόλεων, στο πλήθος των λεωφόρων, και στο πλήθος των οδικών έργων. Σε κάθε μία από τις επόμενες M γραμμές, θα δίνονται τρεις θετικοί ακέραιοι a_i , b_i , και t_i που δηλώνουν ότι υπάρχει μια λεωφόρος που συνδέει τις πόλεις a_i και b_i και κλείνει για συντήρηση τη χρονική στιγμή t_i . Μεταξύ κάθε ζεύγους πόλεων, θα υπάρχει το πολύ μία λεωφόρος που τις συνδέει.

Δεδομένα Εξόδου: Το πρόγραμμα σας πρέπει να τυπώνει στο standard output (στην πρώτη γραμμή) έναν θετικό ακέραιο, τη μέγιστη χρονική στιγμή T^* όπου εξασφαλίζεται ότι όλοι μπορούν να μεταβούν στην εργασία τους, αν κάνουμε τα οδικά έργα με βέλτιστο τρόπο. Αν τα οδικά έργα μπορούν να εξασφαλίσουν απεριόριστο χρόνο ξεκούρασης για τους κατοίκους, το πρόγραμμα πρέπει να τυπώνει μόνο μία γραμμή με τη λέξη “infinity”.

Περιορισμοί:	Παράδειγμα Εισόδου:	Παράδειγμα Εξόδου:
$1 \leq N, M, K \leq 3 \cdot 10^5$	4 5 1	8
$1 \leq t_i \leq 2 \cdot 10^9$	4 2 4	
Όριο χρόνου εκτέλεσης: 1 sec.	3 4 10	
Όριο μνήμης: 64 MB.	4 1 5	
	2 1 3	
	3 2 8	

Άσκηση 2: Σχέδιο Ληστείας

Στην μακρινή Λεφτούπολη, μία τράπεζα διακινεί κάθε μέρα μεγάλα ποσά χρημάτων. Για τις χρηματοπιστολές, χρησιμοποιεί ένα συνεκτικό υποδίκτυο του οδικού δικτύου της πόλης. Προκειμένου να μειώσει τις πιθανότητες ληστείας, κάθε πρωί εφαρμόζει μέτρα ασφαλείας στους δρόμους του υποδικτύου αυτού. Τα μέτρα ασφαλείας έχουν κόστος c_e για κάθε δρόμο e . Για ακόμα μεγαλύτερη ασφάλεια, η τράπεζα αλλάζει καθημερινά το υποδίκτυο που χρησιμοποιεί με βάση έναν σύνθετο αλγόριθμο που καθιστά εξαιρετικά δύσκολη την πρόβλεψη του υποδικτύου. Ωστόσο, ο καινούριος διευθυντής προκειμένου να μειώσει το κόστος αυτής της διαδικασίας, αποφάσισε ότι το δίκτυο που θα επιλέγεται θα είναι υποχρεωτικά ελάχιστου κόστους. Μία συμμορία ληστών, μαθαίνοντας την πληροφορία αυτή, αποφασίζει να αναλύσει το οδικό δίκτυο της πόλης και να προσδιορίσει με ακρίβεια τους πιθανούς δρόμους που θα χρησιμοποιήσει η χρηματοπιστολή.

Η συμμορία σας προσφέρει σημαντική αμοιβή για ένα πρόγραμμα που θα βρίσκει ποιοι δρόμοι θα χρησιμοποιηθούν σίγουρα για χρηματοπιστολές, ποιοι σίγουρα όχι και ποιοι είναι αβέβαιοι.

Λεδομένα Εισόδου: Αρχικά, το πρόγραμμά σας θα διαβάξει από το standard input δύο θετικούς ακέραιους N και M που αντιστοιχούν στο πλήθος των κόμβων και στο πλήθος των δρόμων του δικτύου. Οι κόμβοι του δικτύου αριθμούνται από 1 μέχρι N . Σε κάθε μία από τις επόμενες M γραμμές, θα υπάρχουν τρεις φυσικοί αριθμοί a_e, b_e, c_e χωρισμένοι με κενό. Αυτοί δηλώνουν ότι ο δρόμος $e = \{a_e, b_e\}$ έχει κόστος c_e . Το δίκτυο θεωρείται μη κατευθυνόμενο, θα είναι συνεκτικό και δεν θα περιέχει ανακυκλώσεις ή πολλαπλές ακμές.

Λεδομένα Εξόδου: Το πρόγραμμα σας πρέπει να τυπώνει στο standard output, σε τρεις διαδοχικές γραμμές, τον αριθμό των ακμών που ανήκουν σε κάθε συνεκτικό υποδίκτυο ελάχιστου κόστους, τον αριθμό των ακμών που δεν ανήκουν σε κανένα συνεκτικό υποδίκτυο ελάχιστου κόστους και τον αριθμό των ακμών που ανήκουν σε κάποιο συνεκτικό υποδίκτυο ελάχιστου κόστους, αλλά όχι σε όλα.

Περιορισμοί:	Παραδείγματα Εισόδου:	Παραδείγματα Εξόδου:
$2 \leq N \leq 50.000$	3 3	0
$N - 1 \leq M \leq 500.000$	1 2 1	0
$1 \leq c_e \leq 200$	1 3 1	3
Όριο χρόνου εκτέλεσης: 1 sec.	2 3 1	
Όριο μνήμης: 64 MB.	3 3	2
Στο 60% των αρχείων, θα υπάρχουν το πολύ 500 ακμές ίδιου βάρους.	1 2 1	1
	1 3 1	0
	2 3 2	