Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σουλίου, Θανάσης Λιανέας
3η Σειρά Προγραμματιστικών Ασκήσεων - Ημ/νία Παράδοσης 18/1/2021

Άσκηση 1: Αποστάσεις σε Δίκτυο με Εκθετικά Μήκη Ακμών

Η Επαρχία των Εκθετικών Αποστάσεων είναι μια ενδιαφέρουσα περιοχή της χώρας των Αλγορίθμων. Μεταξύ άλλων, χαρακτηρίζεται από ένα πολύ ιδιαίτερο οδικό δίκτυο. Αποτελείται από N πόλεις που συνδέονται μεταξύ τους με M δρόμους διπλής κατεύθυνσης. Το μήκος κάθε δρόμου είναι μια διαφορετική δύναμη του 2, με τον συντομότερο δρόμο να έχει μήκος 2^0 km και τον μακρύτερο δρόμο να έχει μήκος 2^{M-1} km. Από τότε που μάθατε για το οδικό δίκτυο της Επαρχίας των Εκθετικών Αποστάσεων, δεν μπορείτε να σταματήσετε να το σκέφτεστε! Θέλετε να καταλάβετε τη δομή των συντομότερων διαδρομών μεταξύ των πόλεων, πως αυτές μπορούν να υπολογιστούν αποδοτικά, ποιος είναι ο ευκολότερος τρόπος για να τις αναπαραστήσετε, και πόσες ακμές συνολικά θα χρησιμοποιηθούν.

Ως πρώτο βήμα, θέλετε να γράψετε ένα πρόγραμμα που υπολογίζει το άθροισμα του μήκους των συντομότερων διαδρομών για όλα τα ζευγάρια πόλεων. Αφού το μήκος κάθε δρόμου είναι μια διαφορετική δύναμη του 2, το μήκος κάθε συντομότερης διαδρομής μπορεί να γραφεί φυσιολογικά στο δυαδικό σύστημα. Καλύτερα λοιπόν το πρόγραμμά σας να υπολογίζει την (ενδεχομένως πολύ μεγάλη!) δυαδική αναπαράσταση του συνολικού μήκους των συντομότερων διαδρομών για όλα τα ζευγάρια πόλεων.

Δεδομένα Εισόδου: Το πρόγραμμά σας θα διαβάζει από το standard input δύο θετικούς ακέραιους N και M που αντιστοιχούν στο πλήθος των πόλεων και στο πλήθος των δρόμων του οδικού 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 που δηλώνουν ότι ο δρόμος $e = \{a_e, b_e\}$, που συνδέει τις πόλεις a_e και b_e, έχει μήκος 2^{c_e}. Το οδικό δίκτυο είναι μη κατευθυνόμενο, θα είναι συνεκτικό και δεν θα περιέχει ανακυκλώσεις ή πολλαπλές ακμές. Τα c_e θα είναι διαφορετικά μεταξύ τους.

Δεδομένα Εξόδου: Το πρόγραμμα σας πρέπει να τυπώνει στο standard output, σε μία μόνο γραμμή, τη δυαδική αναπαράσταση του συνολικού μήκους των συντομότερων διαδρομών για όλα τα ζευγάρια πόλεων[1]. Προσέξτε ότι για μεγάλες τιμές των N και M, το αποτέλεσμα μπορεί να έχει μήκος αρκετών χιλιάδων δυαδικών ψηφίων. Ένας τρόπος να αναπαραστήσετε το αποτέλεσμα είναι να διατηρείτε μια ακολουθία $(x_0, c_0), \ldots, (x_k, c_k), \ldots$, όπου το ζεύγος (x_k, c_k) δηλώνει ότι η ακμή μήκους 2^{c_k} περιέχεται x_k φορές στα συντομότερα μονοπάτια μεταξύ των πόλεων. Τελικά, το πρόγραμμά σας πρέπει να τυπώνει τη δυαδική αναπαράσταση του αθροίσματος $\sum_k x_k 2^{c_k}$.

Περιορισμοί:

$3 \leq N \leq 10^5$
$N - 1 \leq M \leq 2 \cdot 10^5$
$c_e \in \{0, 1, \ldots, M-1\}$
Όριο χρόνου εκτέλεσης: 1 sec.
Όριο μνήμης: 64 MB.

Παράδειγμα Εισόδου:

```
5 6
1 3 5
4 5 0
2 1 3
3 2 1
4 3 4
4 2 2
```

Παράδειγμα Εξόδου:

```
1000100
```

[1] **Εξήγηση Παραδείγματος:** Υπολογίζοντας τα συντομότερα μονοπάτια μεταξύ όλων των 10 ζευγαριών πόλεων, βρίσκουμε ότι η ακμή $\{4, 5\}$, μήκους $2^0 = 1$, συμμετέχει σε 4 από αυτά, η ακμή $\{2, 3\}$, μήκους $2^1 = 2$, συμμετέχει επίσης σε 4, η ακμή $\{2, 4\}$, μήκους $2^2 = 4$, συμμετέχει σε 6, και η ακμή $\{1, 2\}$, μήκους $2^3 = 8$, συμμετέχει σε 4. Οι υπόλοιπες ακμές δεν συμμετέχουν σε κανένα συντομότερο μονοπάτι. Το άθροισμα του μήκους των συντομότερων διαδρομών για όλα τα ζευγάρια πόλεων είναι $4 \cdot 1 + 4 \cdot 2 + 6 \cdot 4 + 4 \cdot 8 = 68$, η δυαδική αναπαράσταση του οποίου είναι 1000100.

Άσκηση 2: Δρομολόγια Τρένων

Η πανδημία έχει οδηγήσει πολλές χώρες του κόσμου να λάβουν περιοριστικά μέτρα, με μια μάλλον αναπάντεχη συνέπεια (και) στα δρομολόγια τρένων. Λόγω των μέτρων, κάθε τρένο αναγκάζεται να προσπεράσει κάποιους σταθμούς κατά την διαδρομή του, χωρίς να κάνει στάση σε αυτούς. Συγκεκριμένα, ένα τρένο επιτρέπεται να κάνει στάση σε κάποιον σταθμό s_j, μόνο αν η απόστασή του από κάποια προηγούμενη στάση του τρένου ανήκει σε ένα σύνολο επιτρεπτών αποστάσεων $\{d_1, \ldots, d_N\}$. Υπάρχουν συνολικά 10^9 σταθμοί και η απόσταση μεταξύ κάθε ζεύγους διαδοχικών σταθμών είναι 1. Επομένως, ένα τρένο μπορεί να κάνει στάση σε κάποιον σταθμό s_j, μόνο αν η απόστασή του από την αφετηρία μπορεί να εκφραστεί ως άθροισμα πολλαπλασίων των επιτρεπτών αποστάσεων $\{d_1, \ldots, d_N\}$. Η Σιδηροδρομική Εταιρία έχει αντιληφθεί τις δυσκολίες μετακίνησης που προκαλούν οι περιορισμοί και έχει φροντίσει να υπάρχει τουλάχιστον μια απόσταση $d_k \leq 10^4$ στο σύνολο των επιτρεπτών αποστάσεων, ώστε τα τρένα να μπορούν να κάνουν στάση σχετικά συχνά και να εξυπηρετούνται οι επιβάτες.

Εσείς σπουδάζετε στην πόλη με τον σταθμό 0 και θέλετε να ταξιδέψετε με το τρένο στην ιδιαίτερη πατρίδα σας για τις γιορτές. Επειδή γνωρίζετε ότι αυτό μπορεί να μην είναι εφικτό, έχετε αρχίσει να μελετάτε εναλλακτικές (πιθανόν με κάποιον συνδυασμό μέσων μεταφοράς). Για αυτόν τον λόγο, θέλετε να μπορείτε να απαντάτε ερωτήματα του τύπου "*Μπορώ να φτάσω με το τρένο στον σταθμό s_j αν ξεκινήσω από τον σταθμό 0;*" Συνολικά χρειάζεται να απαντήσετε Q τέτοια ερωτήματα, οπότε προτιμάτε να αυτοματοποιήσετε την διαδικασία απάντησης, για να μην σπαταλάτε χρόνο.

Να γράψετε ένα πρόγραμμα που απαντά στα Q ερωτήματά σας, δοθέντων των N επιτρεπτών αποστάσεων για τις στάσεις. Μπορείτε να θεωρήσετε ότι η πρώτη στάση του τρένου θα είναι στον σταθμό 0, που είναι και η αφετηρία της διαδρομής σας.

Δεδομένα Εισόδου: Το πρόγραμμά σας θα διαβάζει από το standard input δύο θετικούς ακέραιους N και Q, που αντιστοιχούν στο πλήθος των επιτρεπτών αποστάσεων και στο πλήθος Q των ερωτημάτων που θέλετε να απαντήσετε. Στην επόμενη γραμμή θα υπάρχουν N θετικοί ακέραιοι αριθμοί $d_1, \ldots, d_N$ που αντιστοιχούν στις επιτρεπτές αποστάσεις μεταξύ δύο στάσεων. Θα ακολουθούν Q γραμμές. Στην j-οστή από αυτές, θα υπάρχει ένας θετικός ακέραιος s_j που αντιστοιχεί στην απόσταση του σταθμού που σας ενδιαφέρει στην j-οστή ερώτηση από την αφετηρία.

Δεδομένα Εξόδου: Το πρόγραμμα πρέπει να τυπώνει στο standard output συνολικά Q γραμμές. Η j-οστή γραμμή πρέπει να περιέχει τη λέξη "YES", αν το τρένο επιτρέπεται να κάνει στάση στον σταθμό s_j, ή "NO", διαφορετικά[2].

Περιορισμοί:

$1 \leq N \leq 10^3$
$1 \leq Q \leq 10^5$
$1 \leq d_i \leq 10^9$
$1 \leq \min_i\{d_i\} \leq 10^4$
$1 \leq s_j \leq 10^9$
Όριο χρόνου εκτέλεσης: 1 sec.
Όριο μνήμης: 64 MB.

Παράδειγμα Εισόδου:

```
3 3
6 7 5
5
8
13
```

Παράδειγμα Εξόδου:

```
YES
NO
YES
```

Στο 50% των αρχείων εισόδου, θα είναι $\max_i\{d_i\} \leq 10^4$ και $\max_j\{s_j\} \leq 10^4$.

[2] **Εξήγηση Παραδείγματος:** Υπάρχουν 3 επιτρεπτές αποστάσεις, $d_1 = 5, d_2 = 6$ και $d_3 = 7$, και 3 ερωτήματα, $s_1 = 5, s_2 = 8$ και $s_3 = 13$. Οι πρώτες 8 στάσεις του τρένου θα είναι οι $0, 5, 6, 7, 10(= 5+5), 11(= 6+5), 12(= 6+6), 13(= 7+6)$. Οπότε η απάντηση στο πρώτο και το τρίτο ερώτημα είναι "YES", ενώ η απάντηση στο δεύτερο ερώτημα είναι "NO", αφού δεν επιτρέπεται η στάση στον σταθμό 8.