en[i]gma 0x02 Advanced

15 Δεκεμβρίου 2024

Προβλήματα

Πρόβλημα	Χρονικό Όριο	Όριο Μνήμης	Βαθμολογία
Τώρα σκέφτεσαι με πύλες	1 sec	64MB	100
Το καλό τραπεζομάντηλο	1 sec	64MB	100
NIM	1 sec	512MB	100
Πάρτυ έκπληξη	1 sec	64MB	100
Σύνολο			400

Τώρα σκέφτεσαι με πύλες

Βρισκόμαστε σε έναν ουρανοξύστη και θέλουμε να φτάσουμε στον T-οστό όροφο. Ξεκινάμε από τον S-οστό και μπορούμε να χρησιμοποιήσουμε 2 τρόπους να αλλάξουμε όροφο:

- Μέσω **σκαλών**: κάθε σκάλα μας πηγαίνει από έναν όροφο στον αμέσως από πάνω ή στον αμέσως από κάτω.

- μέσω **πυλών**: κάθε όροφος μας προσφέρει (μέχρι) 2 πύλες. Κάθε πύλη μπορεί να μας πάει σε έναν ψηλότερο όροφο ή σε έναν χαμηλότερο όροφο.
  Σε ποιους ορόφους όμως; **Δεν ξέρουμε!** Οι πύλες είναι προκαθορισμένες, αλλά άγνωστες σε εμάς εκ των προτέρων.
  Το μόνο που ξέρουμε είναι ότι:

  – αν χρησιμοποιήσουμε την `επάνω` πύλη του i-οστού ορόφου, δεν υπάρχει καμία μετέπειτα πύλη που φτάνει στον i-οστό όροφο ή παρακάτω από αυτόν και
  – αν χρησιμοποιήσουμε την `κάτω` πύλη, δεν υπάρχει καμία μετέπειτα που φτάνει στον i-οστό όροφο ή παραπάνω από αυτόν άρα, υπάρχει πιθανότητα ένας όροφος να έχει λιγότερες από 2 πύλες.

  Για παράδειγμα, αν ο όροφος 1 έχει πύλη στον όροφο 2 (`επάνω` πύλη), τότε δεν γίνεται ο όροφος 2 να έχει κάτω πύλη κάπου.

Πρέπει να βρείτε τον καλύτερο τρόπο ώστε να φτάσετε από τον όροφο S στον όροφο T.

Η βέλτιστη λύση δεν χρησιμοποιεί καθόλου σκάλες!

Δεδομένα εισόδου (`STDIN`) - εξόδου (`STDOUT`)

Το πρόβλημα είναι διαδραστικό, που σημαίνει ότι δίνετε οδηγίες στον judge και αυτός σας απαντάει με πληροφορίες για να συνεχίσετε.

1. Η πρώτη γραμμή περιέχει 3 ακεραίους F, S και T: το πλήθος των ορόφων, τον όροφο από τον οποίο ξεκινάτε και τον όροφο στον οποίο βρίσκεται η έξοδος. Θεωρείτε πλέον ότι βρίσκεστε στον όροφο S.

2. Όσο δεν βρίσκεστε στον όροφο T:

   - εκτυπώνετε την κίνησή σας: `LADDER UP`, `LADDER DOWN`, `PORTAL UP`, `PORTAL DOWN`
   - ο judge εκτυπώνει με την σειρά του τον αριθμό x του ορόφου στον οποίον καταλήξατε με την μορφή `OK x` και
   - αν ζητήσατε κίνηση `PORTAL` σε όροφο που δεν έχει την αντίστοιχη, ο judge εκτυπωνει `SAME`

3. Μόλις βρεθείτε στον όροφο T, το παιχνίδι τελειώνει και ο judge υπολογίζει το σκορ σας.

Η βαθμολογία για αυτό το παιχνίδι θα είναι: $100\% - \frac{\text{πλήθος κινήσεων σκάλας}}{\text{συνολικές κινήσεις}}$

Παράδειγμα

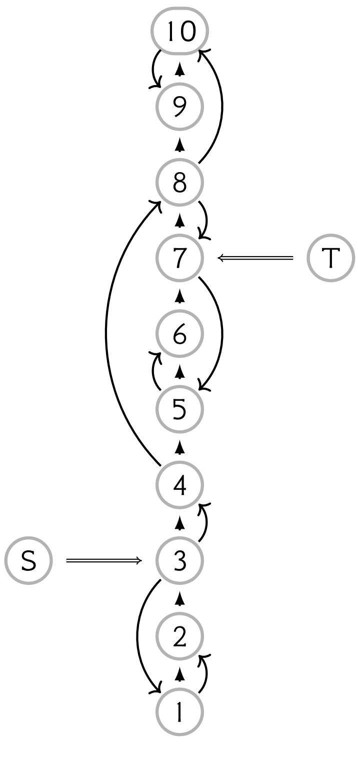
Είσοδος (STDIN)

```
10 3 7
```

Μία πιθανή ακολουθία κινήσεων θα μπορούσε να είναι:

Έξοδος προγράμματος (STDOUT)	Έξοδος βαθμολογητή (STDIN)
PORTAL DOWN	
	OK 1
LADDER UP	
	OK 2
LADDER UP	
	OK 3
PORTAL UP	
	OK 4
PORTAL UP	
	OK 8
LADDER DOWN	
	7

και αυτή η λύση θα βαθμολογηθεί με $\frac{73}{7} = 57\%$.



Σημείωση: για να λειτουργεί σωστά ο διαδραστικός grader, πρέπει να κάνετε flush τις γραμμές που εκτυπώνετε. Αυτό γίνεται ως εξής:

- Στην python: η `print()` το κάνει αυτόματα

```
...
print("LADDER UP")
...
```

- Στην C/C++ (`cstdio`): χρησιμοποιείστε την `printf` κανονικά

```
...
printf("LADDER UP");
...
```

- Στην C++ (`iostream`): χρησιμοποιείστε το `endl`

```
...
cout << "LADDER UP" << endl;
...
```

Το καλό τραπεζομάντηλο

Η μαμά, όταν έρχονται καλεσμένοι, βγάζει πάντα το καλό σερβίτσιο και το καλό τραπεζομάντηλο. Όμως αύριο έχει ξαφνική επίσκεψη και το τραπεζομάντηλο είναι στα άπλυτα! Ζητάει λοιπόν την βοήθειά του Τοτού να χρωματίσει ένα παλιό τραπεζομάντηλο, μπας και σώσει την κατάσταση.

Είναι πολύ αυστηρή στο τι κάνει ένα καλό τραπεζομάντηλο:

- Έχει μόνο 2 χρώματα: κόκκινο και πράσινο
- Έχει οριζόντιες και κατακόρυφες γραμμές που το χωρίζουν σε ορθογώνια (μέρη)
- Για να φαίνονται με το μάτι οι γραμμές, πρέπει τα μέρη που έχουν κοινή ακμή να είναι χρωματισμένα με διαφορετικό χρώμα.

Δίνει το τραπεζομάντηλο στον Τοτό και του υποδεικνύει τις γραμμές που θέλει να φαίνονται. Ο Τοτός πρέπει να το χρωματίσει κατάλληλα.

Δεδομένα εισόδου (`STDIN`)

Στην πρώτη γραμμή δίνονται 2 ακέραιοι θετικοί αριθμοί:

- ο N, το μήκος και

- ο , το πλάτος του τραπεζομάντηλου, χωριζομένοι από έναν χαρκτήρα κενού (`' '`).

Ακολουθεί νέα γραμμή, που ξεκινάει με έναν ακέραιο, θετικό αριθμό h: το πλήθος των οριζόντιων γραμμών, ακολουθούμενο από νέα γραμμή, που περιλαμβάνει h ακέραιους, θετικούς αριθμούς h_i, χωριζομένοι από έναν χαρακτήρα κενού (`' '`), όπου h_i η απόσταση της i-οστής γραμμής από την πάνω πλευρά του τραπεζομάντηλου.

Ακολουθεί νέα γραμμή, που ξεκινάει με έναν ακέραιο, θετικό αριθμό v: το πλήθος των κατακόρυφων γραμμών, ακολουθούμενο από νέα γραμμή, που περιλαμβάνει v ακέραιους, θετικούς αριθμούς v_i, χωριζομένοι από έναν χαρακτήρα κενού (`' '`), όπου v_i: η απόσταση της i-οστής γραμμής από την πιο αριστερή πλευρά του τραπεζομάντηλου.

Ισχύει ότι: $K = h + v$.

Δεδομένα εξόδου (`STDOUT`)

Το πρόγραμμα θα εκτυπώνει στην έξοδο N γραμμές και καθεμία θα έχει M χαρακτήρες που είναι είτε ο `G`, είτε ο `R`, έτσι ώστε τελικά να σχηματίζεται στην έξοδο το τραπεζομάντηλο μετά την παρέμβαση του Τοτού.

Σημείωση: Αν υπάρχουν περισσότερες από 1 πιθανές απαντήσεις, τυπώστε στην έξοδο όποια επιθυμείτε.

Παραδείγματα

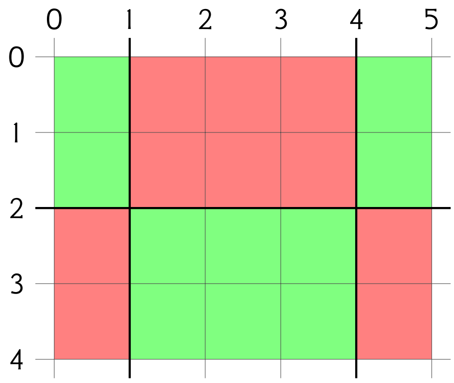
1ο

Είσοδος (STDIN)

```
4 5
1 2
2 1 4
```

Έξοδος (STDOUT)

```
GRRRG
GRRRG
RGGGR
RGGGR
```



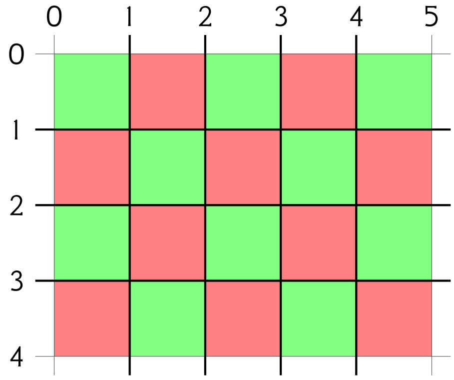
2ο

Είσοδος (STDIN)

```
4 5
3 1 2 3
4 1 2 3 4
```

Έξοδος (STDOUT)

```
GRGRG
RGRGR
GRGRG
RGRGR
```



Υποπροβλήματα

- 70% των πόντων: $1 \leq N, M, K \leq 500$
- 100% των πόντων: $1 \leq N, M \leq 5,000, \quad 1 \leq K \leq 10^6$
  Μπορεί να εμανιστεί πολλές φορές η ίδια γραμμή - στήλη: αν εμφανιστεί περιττό αριθμό φορών πρέπει να εμφανιστεί στο τραπεζομάντηλο, αλλιώς όχι.

NIM

Η Αννούλα και ο Τοτός παίζουν το γνωστό παιχνίδι Νιμ. Το παιχνίδι αυτό παίζεται ως εξής:

> *«Ξεκινώντας από έναν αριθμό βόλων στο τραπέζι, κάθε παίκτης πρέπει να αφαιρέσει 1, 2, ή 3 βόλους στη σειρά του.*
> *Νικητής είναι αυτός που θα πάρει τον τελευταίο βόλο.*
> *Πρώτη ξεκινά πάντα η Αννούλα.»*

Μιας και οι 2 παίκτες παίζουν βέλτιστα, ξέρουν εκ των προτέρων ότι ανάλογα με τον αριθμό των βόλων στην αρχή του παιχνιδιού, το αποτέλεσμα είναι προκαθορισμένο. Έτσι, βαρέθηκαν να παίζουν μόνο με αυτούς τους κανόνες και πλέον πειραματίζονται με διαφορετικές επιλογές στον αριθμό βόλων που μπορούν να αφαιρέσουν.

Βοηθήστε την Αννούλα να μάθει ποιοι αρχικοί αριθμοί βόλων της εγγυούνται ότι, αν παίξει βέλτιστα, τότε θα κερδίσει!

Προσοχή! Αν οι βόλοι που έμειναν είναι λιγότεροι από μια επιλογή, τότε η Αννούλα και ο Τοτός δεν μπορούν να παίξουν εκείνη την κίνηση.

Δεδομένα εισόδου (`STDIN`)

Στην 1η γραμμή δίνεται ένας θετικός, ακέραιος αριθμός M $(M \leq 10)$, το πόσες επιλογές θα έχουν οι παίκτες για να αφαιρέσουν βόλους.
Στην 2η γραμμή δίνονται Μ αριθμοί, οι επιλογές $m_1, m_2, \ldots, m_M (m_i \leq 100)$ σε αύξουσα σειρά.
Στην 3η γραμμή δίνεται ένας θετικός ακέραιος αριθμός Q, το πλήθος των ερωτημάτων για αρχικό αριθμό βόλων.
Ακολουθούν Q $(Q \leq 20)$ γραμμές, που καθεμία έχει γραμμένο έναν αριθμό q_i, τον αρχικό αριθμό βόλων.

Δεδομένα Εξόδου (`STDOUT`)

Η έξοδος αποτελείται από Q γραμμές, με έναν αριθμό σε κάθε γραμμή, `0` ή `1` ανάλογα με το αν η Αννούλα μπορεί **σίγουρα** να κερδίσει με τον αντίστοιχο αρχικό αριθμό βόλων.

Παραδείγματα

1ο

Είσοδος (STDIN)

```
3
1 2 3
5
2
4
5
8
10
```

Έξοδος (STDOUT)

```
1
0
1
0
1
```

Εξήγηση παραδείγματος:

Οι παίκτες έχουν 3 επιλογές.
Μπορούν να αφαιρέσουν 1, 2 ή 3 βόλους κάθε φορά.
Επιπλέον, δίνονται 5 ερωτήματα:

- $q_1 = 2$: Η Αννούλα μπορεί να παίξει τη δεύτερη κίνηση και να κερδίσει απευθείας το παιχνίδι.
- $q_2 = 4$: Όποια κίνηση και να διαλέξει η Αννούλα, ο Τοτός μπορεί να κερδίσει.
  - Αν η Αννούλα αφαιρέσει 1 βόλο, ο Τοτός μπορεί να πάρει τους υπόλοιπους 3.
  - Αν αφαιρέσει 2 βόλους, τότε ο Τοτός πάλι κερδίζει, αφαιρώντας τους άλλους 2.
  - Τέλος, αν αφαιρέσει 3 βόλους, ο Τοτός παίρνει τον τελευταίο 1.
- $q_3 = 5$: Η Αννούλα μπορεί να αφαιρέσει μόνο 1 βόλο και να ρίξει τον Τοτό στην κατάσταση με 4 βόλους, από την οποία, όπως εξηγήθηκε παραπάνω, δεν μπορεί να κερδίσει. Έτσι αφού ο Τοτός χάνει, η Αννούλα κερδίζει.
- $q_4 = 8$: Με όμοια λογική, η Αννούλα δεν μπορεί να κερδίσει ξεκινώντας από τους 8 βόλους.
- $, q = 10$: Με όμοια λογική, η Αννούλα μπορεί να κερδίσει ξεκινώντας από τους 10 βόλους.

2ο

Είσοδος (STDIN)

```
3
1 3 4
3
2
5
7
```

Έξοδος (STDOUT)

```
0
1
0
```

Υποπροβλήματα

- 20% των πόντων: τα m_i είναι οι αριθμοί $\{1, 2, \ldots, M\}$, $q_i \leq 100000$
- 30% των πόντων: $q_i \leq 25$
- 50% των πόντων: $q_i \leq 10000$

Πάρτυ έκπληξη

Οι φίλοι του Τοτού και της Αννούλας έχουν εντυπωσιαστεί από την εργατικότητά τους και αποφασίζουν για να τους χαλαρώσουν να τους οργανώσουν ένα πάρτυ έκπληξη. Αρχίζουν και γράφουν σε μια λίστα τι αντικείμενα χρειάζονται για να γίνει το πάρτυ όσο πιο διασκεδαστικό γίνεται και μια τιμή που δηλώνει τη σημασία αυτού το αντικείμενου για το πάρτυ, δηλαδή το κόστους που θα υπάρξει αν αυτό το αντικείμενο δεν είναι στο πάρτυ.

Οι K φίλοι αποφασίζουν να χωρίσουν μεταξύ τους τα N αντικείμενα της λίστας με τη σειρά, ώστε ο καθένας να αναλάβει να φέρει στο πάρτυ ένα συνεχόμενο τμήμα αντικειμένων από αυτή. Παρόλα αυτά, επειδή αναγνωρίζουν πως υπάρχει πάντα πιθανότητα κάποιος να ξεχάσει να φέρει κάποιο αντικείμενο, και, όσα περισσότερα αντικείμενα αναλάβει να φέρει κάποιος, τόσο πιο πιθανό είναι να ξεχάσει κάποιο από αυτά, οι φίλοι θέλουν να σιγουρευτούν ότι το συνολικό εκτιμώμενο κόστος που μπορεί να υπάρξει θα είναι το ελάχιστο.
Πιο αυστηρά: Δίνονται θετικοί ακέραιοι αριθμοί:

- ο N, που δηλώνει το πλήθος των αντικειμένων
- ο K, που δηλώνει το πλήθος των φίλων και

η ακολουθία A που αποτελείται από N θετικούς ακεραίους $A_1, A_2, \ldots, A_N$, που δηλώνουν το κόστος απουσίας του κάθε αντικειμένου.

Θέλουμε να χωρίσουμε την A σε K τμήματα με τέτοιο τρόπο ώστε να ελαχιστοποιείται το άθροισμα του εκτιμώμενου κόστους κάθε τμήματος. Ως εκτιμώμενο κόστος ενός τμήματος του A που αποτελείται από τις (συνεχόμενες) θέσεις $\{i, i+1, \ldots, j\}$ με $i \leq j$ ορίζεται η ποσότητα:

$$cost(i, j) = \sum_{w=i}^{j} A_w \cdot (j - i + 1)$$

Δηλαδή, θέλουμε να βρούμε την ελάχιστη τιμή της ποσότητας:

$$\text{cost}(1, i_1 - 1) + \text{cost}(i_1, i_2 - 1) + \cdots + \text{cost}(i_{K-1}, N)$$

όπου $1 < i_1 < i_2 < \cdots < i_{K-1} \leq N$.

Δεδομένα εισόδου (STDIN)

Στην 1η γραμμή δίνονται δύο ακεραίοι αριθμοί N και K, χωριζομένοι μεταξύ τους με έναν χαρακτήρα κενού (' '): το πλήθος αντικειμένων και το πλήθως φίλων, αντίστοιχα.
Στην 2η γραμμή δίνονται N ακέραιοι αριθμοί $A_1, \ldots, A_N$: το κόστος απουσίας του κάθε αντικειμένου.

Δεδομένα εξόδου (STDOUT)

Η έξοδος αποτελείται από μία γραμμή με έναν ακέραιο αριθμό, το ελάχιστο συνολικό εκτιμώμενο κόστος.

Παράδειγμα

Είσοδος (STDIN)

```
6 3
11 11 11 24 26 100
```

Έξοδος (STDOUT)

```
299
```

Εξήγηση παραδείγματος:

Ο πρώτος φίλος αναλαμβάνει το τμήμα με τα τρία πρώτα αντικείμενα με εκτιμώμενο κόστος $11 \cdot 3 + 11 \cdot 3 + 11 \cdot 3 = 99$, ο δεύτερος φίλος αναλαμβάνει το τμήμα με τα δύο επόμενα αντικείμενα, με εκτιμώμενο κόστος $24 \cdot 2 + 26 \cdot 2 = 100$, και ο τρίτος φίλος αναλαμβάνει το τμήμα με το τελευταίο αντικείμενο, με εκτιμώμενο κόστος $100 \cdot 1 = 100$. Έτσι, το συνολικό εκτιμώμενο κόστος είναι $99 + 100 + 100 = 299$. Αυτό είναι και το ελάχιστο.

Περιορισμοί:

- $1 \leq K \leq N \leq 2000$
- $1 \leq A_1 \leq 10^9$

Υποπροβλήματα

- 20% των πόντων: $K = 2$.
- 40% των πόντων: $K \leq N \leq 250$
- 20% των πόντων: $A_1 \leq A_2 \leq \cdots \leq A_N$, δηλαδή τα στοιχεία της A είναι σε αύξουσα σειρά.
- 20% των πόντων: χωρίς επιπλέον περιορισμούς.