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

Αλγόριθμοι και Πολυπλοκότητα

Διδάσκοντες: Σ. Ζάχος, Δ. Φωτάκης

2η Σειρά Προγραμματιστικών Ασκήσεων - Ημ/νία Παράδοσης 18/12/2014

Άσκηση 1: Ο Μικρός Τσαγκάρης

Στην μακρινή χώρα των Αλγορίθμων ζει ένα μικρός τσαγκάρης. Στο εργαστήριό του έχει δύο μηχανές παραγωγής παπουτσιών, η μία παράγει μόνο δεξιά παπούτσια και η άλλη μόνο αριστερά. Κάθε τύπος δεξιού παπουτσιού μπορεί να γίνει ζευγάρι με ένα μόνο τύπο αριστερού παπουτσιού και αντίστροφα. Λόγω περιορισμένου χώρου στο εργαστήριο, ο τσαγκάρης δεν μπορεί να αποθηκεύει τα παραγόμενα παπούτσια. Έτσι, παίρνοντας ένα παπούτσι από κάθε μηχανή, ο τσαγκάρης έχει τρεις επιλογές:

- Να πετάξει το δεξί παπούτσι και να πάρει ένα επόμενο από τη μηχανή που παράγει δεξιά παπούτσια, προσπαθώντας να το ταιριάξει με το αριστερό παπούτσι που έχει στα χέρια του.
- Να πετάξει το αριστερό παπούτσι και να πάρει ένα επόμενο από τη μηχανή που παράγει αριστερά παπούτσια, προσπαθώντας να το ταιριάξει με το δεξί παπούτσι που έχει στα χέρια του.
- Αν τα παπούτσια είναι ίδιου τύπου, να τα πουλήσει ως ζευγάρι.

Ο τσαγκάρης γνωρίζει την ακολουθία τύπων που παράγει κάθε μηχανή, αλλά δεν μπορεί να την ελέγξει. Μάλιστα γνωρίζει ότι κάθε μηχανή παράγει συνήθως πολλά παπούτσια του ίδιου τύπου διαδοχικά. Γράψτε ένα πρόγραμμα που θα βοηθήσει τον μικρό τσαγκάρη να σχεδιάσει τη στρατηγική του, ώστε να πουλήσει όσο περισσότερα ζευγάρια γίνεται.

Δεδομένα Εισόδου: Αρχικά, το πρόγραμμα θα διαβάζει από το standard input δύο θετικούς ακέραιους L και R που αντιπροσωπεύουν τα μήκη των ακολουθιών τύπων παπουτσιών που παράγουν οι δύο μηχανές. Στη συνέχεια, το πρόγραμμα θα διαβάζει μία γραμμή με $2L$ θετικούς ακεραίους $a_1, A_1, a_2, A_2, \ldots, a_L, A_L$ και άλλη μία γραμμή με $2R$ θετικούς ακεραίους $b_1, B_1, b_2, B_2, \ldots, b_R, B_R$. Οι αριθμοί αυτοί δηλώνουν ότι η μηχανή αριστερών παπουτσιών θα παράξει a_1 παπούτσια τύπου A_1, στη συνέχεια a_2 παπούτσια τύπου A_2, κοκ., μέχρι να σταματήσει παράγοντας a_L αριστερά παπούτσια τύπου A_L, και ότι η μηχανή δεξιών παπουτσιών θα παράξει b_1 παπούτσια τύπου B_1, στη συνέχεια b_2 παπούτσια τύπου B_2, κοκ., μέχρι να σταματήσει παράγοντας b_R δεξιά παπούτσια τύπου B_R. Ένα δεξί και ένα αριστερό παπούτσι, μπορούν να γίνουν ζευγάρι μόνο εάν έχουν τον ίδιο αριθμό τύπου.

Δεδομένα Εξόδου: Το πρόγραμμα πρέπει να τυπώνει στο standard output (στην πρώτη γραμμή) το μέγιστο πλήθος ζευγαριών που μπορούν να πουληθούν. Σημειώστε ότι για μεγάλες τιμές των a_i και b_i, το μέγιστο πλήθος ζευγαριών (καθώς και κάποια από τα ενδιάμεσα αποτελέσματα που χρειάζονται για τον υπολογισμό του) μπορεί να υπερβαίνουν το 2^{32}.

Περιορισμοί:

$1 \leq L, R \leq 100$
$1 \leq A_i, B_i \leq 100$
Στο 60% των test cases θα έχουμε $1 \leq a_i, b_i \leq 25$
Στο υπόλοιπο 40% των test cases θα έχουμε $1 \leq a_i, b_i \leq 10^{16}$
Όριο χρόνου εκτέλεσης: 1 sec.
Όριο μνήμης: 64 MB.

Παράδειγμα Εισόδου:

```
3 5
10 1 6 2 10 1
5 1 3 2 10 1 3 2 5 1

3 5
10 1 6 2 10 1
5 1 6 2 10 1 6 2 5 1

1 1
5000 10
50000 100
```

Παράδειγμα Εξόδου:

```
20

21

0
```

Άσκηση 2: Παιχνίδι Δράσης

Μόλις ξεκινήσατε να παίζετε το νέο multiplayer RPG Algorithmic Quest III, μαζί με τον καλύτερό σας φίλο. Για να έχετε μεγαλύτερες πιθανότητες επιβίωσης αποφασίσατε να δημιουργήσετε μία φατρία.

Για την εκλογή του αρχηγού της φατρίας, η φατρία μπαίνει σε ένα μπουντρούμι που αποτελείται από ένα ορθογώνιο πλέγμα $N \times M$ δωματίων. Σε κάθε δωμάτιο με συντεταγμένες (x, y) βρίσκονται κάποια τέρατα από τα οποία μπορείτε να συλλέξετε $c(x, y)$ χρυσά νομίσματα, ως λάφυρα, μόλις τα εξοντώσετε. Η φατρία μπαίνει στο μπουντρούμι από το δωμάτιο με συντεταγμένες $(1, 1)$, όπου δεν υπάρχουν τέρατα (ούτε νομίσματα), και η συλλογή νομισμάτων ολοκληρώνεται μόλις η φατρία φτάσει στο δωμάτιο (N, M).

Οι δύο υποψήφιοι αρχηγοί της φατρίας αποφασίζουν εναλλάξ, ξεκινώντας από εσάς, ποιο θα είναι το επόμενο δωμάτιο που θα μεταφερθεί η φατρία. Η μεταφορά μπορεί να γίνει σε οποιοδήποτε δωμάτιο αρκεί αυτό να βρίσκεται έως και K δωμάτια δεξιά και έως και K δωμάτια κάτω από το τρέχον δωμάτιο. Έτσι, αν η φατρία βρίσκεται στο δωμάτιο με συντεταγμένες (x, y), το παιχνίδι μπορεί να μεταφερθεί σε οποιοδήποτε δωμάτιο με συντεταγμένες $(x + i, y + j)$, με $0 \leq i \leq K$, $0 \leq j \leq K$, και $i + j > 0$. Για παράδειγμα, αν οι παίκτες βρίσκονται στο δωμάτιο με συντεταγμένες $(3, 4)$ και έχουμε $K = 2$, τότε η φατρία μπορεί να μεταφερθεί σε οποιοδήποτε δωμάτιο από τα $(3, 5)$, $(3, 6)$, $(4, 4)$, $(4, 5)$, $(4, 6)$, $(5, 4)$, $(5, 5)$ και $(5, 6)$. Μετά από την εξόντωση των τεράτων του δωματίου, τα λάφυρα (χρυσά νομίσματα) κερδίζονται από τον υποψήφιο αρχηγό που επέλεξε το δωμάτιο. Αρχηγός της φατρίας εκλέγεται ο παίκτης που κέρδισε τα περισσότερα χρυσά νομίσματα.

Τόσο εσείς όσο και ο φίλος σας έχετε στην διάθεση σας τον χάρτη του μπουντρουμιού, όπου αναγράφεται για κάθε δωμάτιο πόσα είναι τα συνολικά λάφυρα. Στόχος σας είναι να βρείτε μια στρατηγική για την επιλογή των δωματίων σας εξασφαλίζει όσο το δυνατόν περισσότερα χρυσά νομίσματα στο τέλος του παιχνιδιού, δεδομένου ότι και ο φίλος σας εφαρμόζει βέλτιστη στρατηγική. Αν υπάρχουν περισσότερες από μία διαδρομές που να δίνουν μέγιστο συνολικό κέρδος, θα θέλατε να επιλέξετε αυτή που ελαχιστοποιεί το κέρδος του φίλου σας, ώστε να καταδείξετε την υπεροχή σας.

Δεδομένα Εισόδου: Αρχικά, το πρόγραμμα θα διαβάζει από το standard input τρεις θετικούς ακέραιους αριθμούς που αντιστοιχούν στις διαστάσεις του μπουντρουμιού, N και M, και στο μέγιστο βήμα K. Σε κάθε μία από τις επόμενες N γραμμές, θα υπάρχουν M φυσικοί αριθμοί (χωρισμένοι με κενό) που δηλώνουν το πλήθος των χρυσών νομισμάτων που μπορείτε να συλλέξετε από τα δωμάτια με τις αντίστοιχες συντεταγμένες.

Δεδομένα Εξόδου: Το πρόγραμμα πρέπει να τυπώνει στο standard output (στην πρώτη γραμμή) το μέγιστο πλήθος χρυσών νομισμάτων που συλλέγετε εσείς και ο φίλος σας (με αυτή τη σειρά, οι δύο αριθμοί πρέπει να χωρίζονται με ένα κενό) δεδομένου ότι αμφότεροι εφαρμόζετε βέλτιστη στρατηγική.

Περιορισμοί:

$2 \leq N, M \leq 500$
$1 \leq K \leq 10$
$0 \leq c_i \leq 40000$

Όριο χρόνου εκτέλεσης: 1 sec.
Όριο μνήμης: 64 MB.

Bonus: κάποια αρχεία με
$1 \leq N, M \leq 1000$ και
$1 \leq K \leq 200$

Παράδειγμα Εισόδου:

```
4 4 1
0 8 1 1
8 5 2 1
1 1 1 5
2 1 1 4
```

Παράδειγμα Εξόδου:

```
14 10
```