

Camp Προετοιμασίας

Πανελλήνιος Διαγωνισμός Πληροφορικής 2018

Πρόβλημα

ICPC (Senior)

Οι φοιτητές της σχολής Ηλεκτρολόγων Μηχανικών ετοιμάζονται να διαγωνιστούν στον προκριματικό διαγωνισμό για τον ICPC (International Collegiate Programming Contest) ο οποίος διεξάγεται στα νέα κτήρια του Εθνικού Μετσόβιου Πολυτεχνείου. Τη μέρα του διαγωνισμού, οι διαγωνιζόμενοι μαζεύονται στην αίθουσα του εργαστηρίου που περιέχει δύο σειρές από N υπολογιστές η καθεμία. Συνολικά έχουμε $2 \times N$ διαγωνιζόμενους και σκοπός μας είναι να μεγιστοποιήσουμε την συνολική απόδοση της αίθουσας ταιριάζοντας σε ζευγάρια αυτούς που είναι καλοί στον προγραμματισμό με αυτούς που είναι καλοί στους αλγορίθμους. Βάζουμε λοιπόν αυτούς που τα καταφέρνουν καλύτερα στον προγραμματισμό στην πρώτη σειρά υπολογιστών και αυτούς που ειδικεύονται στους αλγορίθμους στη δεύτερη.

Για κάθε διαγωνιζόμενο της πρώτης σειράς, γνωρίζουμε το ταλέντο του στον προγραμματισμό $A[i]$, το οποίο δίνεται σαν ποσοστό επί τις χιλιάδες. Αντίστοιχα, για κάθε άτομο της δεύτερης σειράς γνωρίζουμε το ταλέντο του στους αλγορίθμους $B[i]$ σαν ποσοστό επί τις χιλιάδες. Έτσι, αν κάποιος έχει ταλέντο $A[i] = 999$, αυτό σημαίνει πως είναι 99.9% ταλαντούχος στον προγραμματισμό.

Αν ταιριάξουμε τον i -οστό διαγωνιζόμενο της πρώτης σειράς με τον j -οστό διαγωνιζόμενο της δεύτερης, επιφέρουμε στην απόδοση της αίθουσας όφελος ίσο με $A[i] * B[j]$. Επιπλέον, απαιτούμε τα ζευγάρια που συνεργάζονται να μην “τέμνονται”, δηλαδή αν αποφασίσουμε να ταιριάξουμε δυο ζευγάρια (i_1, j_1) και (i_2, j_2) με $i_1 < i_2$ τότε θα πρέπει και $j_1 < j_2$.

Ωστόσο, πρέπει να επιλέξουμε τα ζευγάρια με προσοχή, καθώς αν αγνοήσουμε κάποιους διαγωνιζόμενους αυτοί θα δυσανεστηθούν και θα συνωμοτήσουν εναντίον των υπολοίπων χρησιμοποιώντας τα ταλέντα τους στην πληροφορική. Συγκεκριμένα, αν αγνοήσουμε τους διαγωνιζόμενους $\{i, i+1, i+2, \dots, j\}$ της πρώτης γραμμής αυτοί θα συνεργαστούν με τους διπλανούς τους και θα επιφέρουν κόστος ίσο με

$$(A[i] + A[i+1] + \dots + A[j])^2$$

το οποίο αφαιρείται απ’ το συνολικό όφελος της αίθουσας. Αντίστοιχα, αν αγνοήσουμε τους διαγωνιζόμενους $\{i, i+1, i+2, \dots, j\}$ της δεύτερης γραμμής το κόστος για την αίθουσα θα είναι ίσο με

$$(B[i] + B[i+1] + \dots + B[j])^2$$

Δεδομένα εισόδου (icpc.in)

Η πρώτη γραμμή της εισόδου θα περιέχει έναν φυσικό αριθμό N , το μισό του πλήθους των διαγωνιζόμενων. Οι επόμενες N γραμμές θα περιέχουν έναν θετικό ακέραιο η καθεμία, το ταλέντο $A[i]$ του i -οστού διαγωνιζόμενου της πρώτης σειράς. Οι επόμενες N γραμμές θα περιέχουν από έναν θετικό ακέραιο η καθεμία, το ταλέντο $B[i]$, το ταλέντο του i -οστού διαγωνιζόμενου της δεύτερης σειράς

Δεδομένα εξόδου (icpc.out)

Η έξοδος πρέπει να περιέχει μία γραμμή με ακριβώς ένα φυσικό αριθμό: Το μέγιστο συνολικό όφελος που μπορούμε να πετύχουμε για την αίθουσα.

Παράδειγμα εισόδου

3
1
1
5
5
1
1

Παράδειγμα εξόδου

17

Εξήγηση: Συνολικά έχουμε 6 διαγωνιζόμενους, τρεις καλούς στον προγραμματισμό με ταλέντα $A[1]=1$, $A[2]=1$, $A[3]=5$ και τρεις καλούς στους αλγορίθμους με ταλέντα $B[1]=5$, $B[2]=1$, $B[3]=1$ αντίστοιχα. Το βέλτιστο συνολικό όφελος προκύπτει αν ταιριάξουμε σε μια ομάδα τον 3ο διαγωνιζόμενο της πρώτης σειράς με τον 1ο διαγωνιζόμενο της δεύτερης. Το συνολικό όφελος από αυτό το ταίριασμα είναι 25, ωστόσο από αυτό αφαιρείται το κόστος από την δυσαρέσκεια του 1ου και του 2ου διαγωνιζόμενου της πρώτης σειράς, καθώς και του 2ου και του 3ου διαγωνιζόμενου της δεύτερης σειράς. Συνολικά έχουμε επομένως:

$$A[3]*B[1] - (A[1] + A[2])^2 - (B[2] + B[3])^2 = 5*5 - (1+1)^2 - (1+1)^2 = 25 - 4 - 4 = 17$$

Περιορισμοί

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

Subtasks

- Σε testcases που θα αντιστοιχούν στο 10% της βαθμολογίας, θα είναι $N \leq 10$
- Σε testcases που θα αντιστοιχούν στο 30% της βαθμολογίας, θα είναι $N \leq 70$
- Σε testcases που θα αντιστοιχούν στο 60% της βαθμολογίας, θα είναι $N \leq 200$
- Σε testcases που θα αντιστοιχούν στο 80% της βαθμολογίας, θα είναι $N \leq 1.000$
- Σε testcases που θα αντιστοιχούν στο 100% της βαθμολογίας, θα είναι $N \leq 2.000$