



Με τη συνεργασία: Αριστοτελείου Παν/μίου Θεσσαλονίκης, Εθνικού & Καποδιστριακού Παν/μίου Αθηνών, Εθνικού Μετσοβείου Πολυτεχνείου, Παν/μίου Αιγαίου, Παν/μίου Ιωαννίνων, Παν/μίου Πατρών, Παν/μίου Πειραιά, Α-ΤΕΙ Αθήνας.

19ος ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕΛΙΚΗ ΦΑΣΗ

Θέμα 1ο: ΕΡΓΑΣΤΗΡΙΟ ΚΒΑΝΤΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ

[Μονάδες 25]

Οι κβαντικοί υπολογιστές λειτουργούν στο επίπεδο των υποατομικών σωματιδίων, όπου ένα κβαντικό δυαδικό ψηφίο μπορεί να θεωρηθεί ότι αντιπροσωπεύει το 1 και το 0 ταυτόχρονα! Κατά τη μέτρηση όμως ή την παρατήρηση, αίρεται η 'ασάφεια'. Για αυτό, ένας κβαντικός υπολογιστής μπορεί θεωρητικά να κάνει πολλές πράξεις ταυτόχρονα. Οι κβαντικοί υπολογιστές χρησιμοποιούν ιδιότητες όπως η μαγνητική ιδιοπεριστροφή (spin) των ατομικών πυρήνων για να αντιπροσωπεύσουν τα κβαντικά δυαδικά ψηφία, ή όπως λέγονται "qubits". Για την ενταμίευση qubits χρησιμοποιούνται κυρίως οργανικές ενώσεις σε κατάλληλες υποδοχές. Με τη βοήθεια τέτοιων ενώσεων, μπορούμε να δημιουργήσουμε συστάδες qubits με μέγεθος από μερικά μέχρι μερικές δεκάδες qubits.

Στο Εργαστήριο Κβαντικών Υπολογιστών του Πανεπιστημίου Αθηνών, οι πειραματικοί υπολογιστές διαθέτουν τρεις υποδοχές qubits (A, B, C). Για να μεταφερθεί ένα σύνολο συστάδων qubit από το A στο C θα πρέπει: Να μεταφέρεται πρώτα η συστάδα με το μεγαλύτερο μέγεθος και μετά αυτή με το μικρότερο. Μπορεί να γίνει μόνο μια μεταφορά μίας συστάδας κάθε φορά και μόνο μέσω των τριών υποδοχών.

Παράδειγμα με 2 qubits :

	A	B	C
	A	B	C
	A	B	C
	A	B	C

Πρόβλημα: Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI, το οποίο θα καθοδηγεί τους επιστήμονες ποια qubits να μετακινούν σε ποια υποδοχή. Πάντα ξεκινούμε από το A για να φθάσουμε στο C μέσω του B. Υποθέστε ότι η υποδοχή A έχει γεμίσει κανονικά με qubits και ότι τα qubits πρέπει να διαταχθούν σωστά και στο C. Σε κάθε περίπτωση πρέπει να γίνουν οι λιγότερες δυνατές μετακινήσεις.

ΔΕΔΟΜΕΝΑ ΕΙΣΟΔΟΥ

Το αρχείο qubits.in έχει ακριβώς έναν ακέραιο αριθμό που δηλώνει τον αριθμό N των συστάδων qubits που περιέχει, όπου $3 \leq N \leq 20$.



Με τη συνεργασία: Αριστοτελείου Παν/μίου Θεσσαλονίκης, Εθνικού & Καποδιστριακού Παν/μίου Αθηνών, Εθνικού Μετσοβείου Πολυτεχνείου, Παν/μίου Αιγαίου, Παν/μίου Ιωαννίνων, Παν/μίου Πατρών, Παν/μίου Πειραιά, Α-ΤΕΙ Αθήνας.

ΔΕΔΟΜΕΝΑ ΕΞΟΔΟΥ

Το αρχείο qubits.out έχει ανά γραμμή ζεύγη χαρακτήρων A B, A C, B C, B A, C A, C B με κενό μεταξύ τους που εκφράζουν τις μετακινήσεις qubits μεταξύ των υποδοχών.

ΠΑΡΑΔΕΙΓΜΑ ΔΕΔΟΜΕΝΩΝ ΕΙΣΟΔΟΥ - ΕΞΟΔΟΥ

qubits.in	qubits.out
3	A C A B C B A C B A B C A C

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα newline χαρακτήρα.
Μέγιστος χρόνος εκτέλεσης: 15 sec.

Θέμα 2ο: ΜΠΑΛΕΣ ΥΑΛΟΥ

[Μονάδες 35]

Η εταιρεία “γιούλα” κατασκευάζει γυάλινες μπάλες από ειδικό γυαλί. Κάθε μέρα παράγει ένα σύνολο από μπάλες ίδιας αντοχής. Ωστόσο, διαφορετικά σύνολα μπορεί να έχουν διαφορετική αντοχή. Για να μετρήσει την αντοχή κάθε συνόλου, ο μηχανικός παραγωγής ακολουθεί την εξής διαδικασία: Αφήνει σε ελεύθερη πτώση κάποιες μπάλες (από ένα σύνολο) από διαφορετικά επίπεδα ενός κτιρίου και, επομένως, παριστάνει την αντοχή τους με το χαμηλότερο επίπεδο στο οποίο η μπάλα θα σπάσει στην ελεύθερη πτώση. Κάθε επίπεδο καθορίζεται από ένα όροφο.

Στην αρχή αποφασίζει να ανεβαίνει ένα-ένα τα πατώματα και να εκτελεί το πείραμα διαδοχικά. Επομένως αν μια μπάλα έχει αντοχή 10, θα εκτελέσει 10 προσπάθειες. Μετά αποφάσισε να κάνει κάτι πιο γρήγορο, με κόστος να χρησιμοποιεί περισσότερες μπάλες. Έτσι αποφάσισε να εκτελεί τυχαία πειράματα από διαφορετικούς ορόφους ώστε να περιορίζει τους ορόφους που θα κάνει προσπάθειες. Το αφεντικό του όμως βρίσκει ότι σπάει πολλές μπάλες έτσι και του προτείνει να βρίσκει την αντοχή χρησιμοποιώντας μόνο δύο μπάλες από κάθε σύνολο.

Μετά από σκέψη ο μηχανικός αποφασίζει να κάνει το εξής. Θα χρησιμοποιήσει την πρώτη μπάλα για μια προσπάθεια από τον μεσαίο όροφο. Αν η μπάλα σπάσει, είναι σίγουρο ότι η αντοχή της είναι μικρότερη ή ίση από αυτή που αντιστοιχεί στο μεσαίο επίπεδο. Επομένως, θα χρησιμοποιήσει την δεύτερη μπάλα για να ελέγξει την αντοχή σειριακά ξεκινώντας από το πρώτο επίπεδο και δοκιμάζοντας διαδοχικά κάθε επίπεδο. Αν όμως δεν σπάσει όταν αφηθεί σε ελεύθερη πτώση από το μεσαίο επίπεδο, τότε θα αποκλείσει το κάτω μισό του κτιρίου και θα συνεχίσει με τον ίδιο τρόπο για το άνω μισό. Δηλαδή χρησιμοποιώντας πάλι την πρώτη μπάλα θα προσπαθήσει να βρει την αντοχή της συνεχίζοντας από το μεσαίο επίπεδο του πάνω μισού του κτιρίου. Παρόμοια, θα εκτελεί ένα βήμα τύπου δυαδικής αναζήτησης αν η πρώτη μπάλα



Με τη συνεργασία: Αριστοτελείου Παν/μίου Θεσσαλονίκης, Εθνικού & Καποδιστριακού Παν/μίου Αθηνών, Εθνικού Μετσοβείου Πολυτεχνείου, Παν/μίου Αιγαίου, Παν/μίου Ιωαννίνων, Παν/μίου Πατρών, Παν/μίου Πειραιά, Α-ΤΕΙ Αθήνας.

αντέχει στην ελεύθερη πτώση ή ένα βήμα γραμμικής αναζήτησης μετά από το σπάσιμο της πρώτης μπάλας.

Πρόβλημα: Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI, που να υπολογίζει την αντοχή ενός συνόλου από μπάλες με τον μικρότερο αριθμό προσπαθειών. Καμιά μπάλα δεν σπάει στο ισόγειο.

ΔΕΔΟΜΕΝΑ ΕΙΣΟΔΟΥ

Το αρχείο `balls.in` έχει στην πρώτη γραμμή έναν ακέραιο L (όπου $2 \leq L \leq 10$) που παριστάνει το πλήθος των συνόλων που θα ελεγχθούν. Κάθε μια από τις L γραμμές που ακολουθούν περιέχει δύο ακέραιους N, M (όπου $1 \leq M \leq N \leq 500$) που αντιπροσωπεύουν τα επίπεδα του κτιρίου (χωρίς το ισόγειο) και την αντοχή της μπάλας, αντίστοιχα.

ΔΕΔΟΜΕΝΑ ΕΞΟΔΟΥ

Το πρόγραμμά σας θα πρέπει να γράφει στο αρχείο `balls.out` L γραμμές. Κάθε γραμμή θα έχει έναν ακέραιο που δηλώνει το ελάχιστο πλήθος ελέγχων που πρέπει να γίνουν για κάθε γραμμή της εισόδου.

ΠΑΡΑΔΕΙΓΜΑ ΔΕΔΟΜΕΝΩΝ ΕΙΣΟΔΟΥ – ΕΞΟΔΟΥ

Παράδειγμα

<code>balls.in</code>	<code>balls.out</code>
2	3
7 2	3
7 5	

Μορφοποίηση: Τόσο στην είσοδο όσο και στην έξοδο, οι αριθμοί σε μια γραμμή χωρίζονται με ένα κενό και όλες οι γραμμές τερματίζουν με ένα newline χαρακτήρα.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Θέμα 3ο: ΘΕΑ ΣΤΗ ΘΑΛΑΣΣΑ

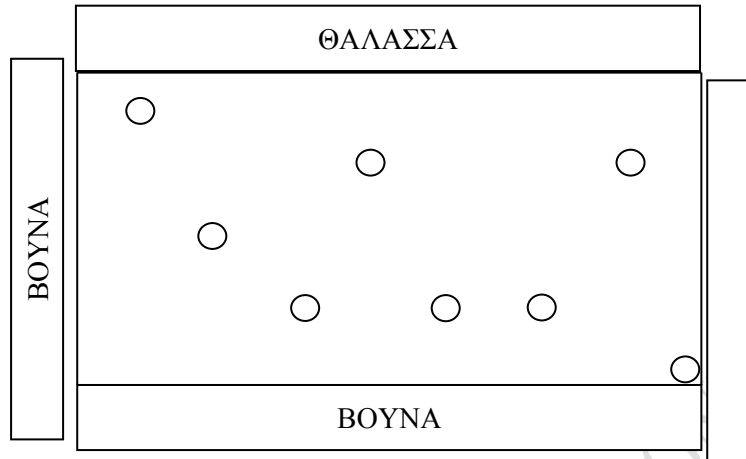
[40 Μονάδες]

Πολλές Ελληνικές πόλεις έχουν θέα στη θάλασσα. Οι τιμές των σπιτιών στις πόλεις αυτές είναι ανάλογες της θέας προς τη θάλασσα. Υποθέστε ότι μια πόλη σχηματίζει ένα ορθογώνιο παραλληλόγραμμο με τη δυτική και τη νότια πλευρά να έχουν θέα στα βουνά (όπως το διάγραμμα). Κάθε σπίτι προσδιορίζεται από τις συντεταγμένες x και y . Για να έχει ένα σπίτι $hi(x_i, y_i)$ καλή θέα δεν πρέπει να υπάρχει άλλο σπίτι $hj(x_j, y_j)$ ώστε να ισχύει: και . Αυτή είναι και η μόνη παράμετρος που επηρεάζει την τιμή των σπιτιών. Δεν υπάρχουν σπίτια που να έχουν την ίδια x συντεταγμένη και το πλήθος των σπιτιών είναι D , που είναι δύναμη του 2.

Πρόβλημα: Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI, που να βρίσκει τις συντεταγμένες των σπιτιών με την υψηλότερη τιμή δηλαδή αυτά με την καλύτερη θέα στη θάλασσα, που βρίσκονται στο βόρειο και ανατολικό τμήμα της πόλης.



Με τη συνεργασία: Αριστοτελείου Παν/μίου Θεσσαλονίκης, Εθνικού & Καποδιστριακού Παν/μίου Αθηνών, Εθνικού Μετσοβείου Πολυτεχνείου, Παν/μίου Αιγαίου, Παν/μίου Ιωαννίνων, Παν/μίου Πατρών, Παν/μίου Πειραιά, Α-ΤΕΙ Αθήνας.



ΔΕΛΟΜΕΝΑ ΕΙΣΟΔΟΥ

Το αρχείο `seaview.in` στη πρώτη γραμμή περιέχει έναν ακέραιο D ($8 \leq D \leq 9000000$) το πλήθος των σπιτιών. Κάθε μια από τις επόμενες D γραμμές περιέχει δύο ακεραίους, που παριστάνουν τις x και y συντεταγμένες των σπιτιών αντίστοιχα, ($0 \leq x, y \leq 14000000$).

ΔΕΛΟΜΕΝΑ ΕΞΟΔΟΥ

Το αρχείο `seaview.out` αποτελείται από L γραμμές, όπου το L αντιπροσωπεύει το πλήθος των σπιτιών με τις υψηλότερες τιμές. Κάθε γραμμή θα περιέχει δύο ακεραίους x και y τις συντεταγμένες των σπιτιών με την υψηλότερη τιμή. Οι L γραμμές θα πρέπει να εμφανίζονται με αύξουσα σειρά σε σχέση με την x συντεταγμένη.

ΠΑΡΑΔΕΙΓΜΑ ΔΕΛΟΜΕΝΩΝ ΕΙΣΟΔΟΥ – ΕΞΟΔΟΥ

<code>seaview.in</code>	<code>seaview.out</code>
8	1 4
5 1	4 3
1 4	7 3
6 1	8 0
3 1	
4 3	
2 2	
8 0	
7 3	

Μορφοποίηση: Τόσο στην είσοδο όσο και στην έξοδο, οι αριθμοί σε μια γραμμή χωρίζονται με ένα κενό και όλες οι γραμμές τερματίζουν με ένα newline χαρακτήρα.

Μέγιστος Χρόνος Εκτέλεσης 13 sec.

Όριο μνήμης: 200 Mbytes