

Όνομα Άσκησης: Mixing Milk

Πηγή: USACO 1.3

Εκφώνηση

Αναμειγνύοντας Γάλα

Επειδή η επιχείρηση συσκευασίας γάλακτος είναι δεν μπορεί να έχει πολύ ψηλό κέρδος είναι βασικό να διατηρηθεί η τιμή του ακατέργαστου γάλακτος όσο το δυνατόν χαμηλότερη. Βοηθήστε την εταιρία Merry Milk Makers να αγοράσει το ακατέργαστο γάλα στη χαμηλότερη τιμή.

Η εταιρία Merry Milk Makers μπορεί να αγοράσει το γάλα που χρειάζεται από διάφορους αγελαδοτρόφους, ο καθένας από τους οποίους έχει τη δική του τιμή στην οποία πουλά το γάλα. Επίσης κάθε αγελαδοτρόφος παράγει συγκεκριμένη ποσότητα για να διαθέσει στην αγορά.

Με δεδομένη την ημερήσια ανάγκη της εταιρίας Merry Milk Makers σε γάλα καθώς και την τιμή ανά γαλόνι και την ποσότητα που παράγει κάθε αγελαδοτρόφος, υπολογίστε το ελάχιστο ποσό που πρέπει να πληρώσει η εταιρία.

Σημείωση: Το γάλα που παράγουν οι αγελαδοτρόφοι είναι αρκετό για να ικανοποιηθεί η ανάγκη της εταιρίας.

Είσοδος

Γραμμή 1:	Δύο ακέραιοι, N και M . Η πρώτη τιμή, N , ($0 \leq N \leq 2,000,000$) είναι η ποσότητα του γάλακτος που χρειάζεται η εταιρία Merry Milk Makers. Η δεύτερη, M , ($0 \leq M \leq 5,000$) αντιστοιχεί στον αριθμό των αγελαδοτρόφων από όπου μπορεί η εταιρία να αγοράσει γάλα.
Γραμμή 2.. $M+1$:	Οι επόμενες M γραμμές περιέχουν 2 ακέραιους, P_i και A_i . P_i ($0 \leq P_i \leq 1,000$) η τιμή σε cents που χρεώνει ο αγελαδοτρόφος i . A_i ($0 \leq A_i \leq 2,000,000$) η ποσότητα που μπορεί ο αγελαδοτρόφος i να πουλήσει στην εταιρία Merry Milk Makers κάθε μέρα.

Παράδειγμα Εισόδου (file milk.in)

```
100 5
5 20
9 40
3 10
8 80
6 30
```

Έξοδος

Ένας ακέραιος αριθμός που αντιστοιχεί στην ελάχιστη τιμή που πρέπει να πληρώσει η εταιρία

Παράδειγμα εξόδου (file milk.out)

630

Hints

Η απληστία κάνει καλό στην εταιρία!

Λύση

Η άσκηση είναι αρκετά εύκολη και αποτελεί απλή εφαρμογή της άπληστης (greedy) τεχνικής. Σε αυτή την τεχνική παίρνουμε πάντα το καλύτερο «κομμάτι» που υπάρχει διαθέσιμο.

Στην αρχή αποθηκεύουμε τα στοιχεία σε ένα δισδιάστατο πίνακα ή σε ένα δυσδιάστατο vector

```
vector< pair<int, int> > a;
```

Προτιμούμε τα vector γιατί δε χρειάζεται να δεσμεύσουμε όλη τη μνήμη από την αρχή. Στη συνέχεια ταξινομούμε τα στοιχεία σε αύξουσα σειρά με βάση τη τιμή. Τέλος έχουμε μια δομή επανάληψης για να πάρουμε τα λίτρα που χρειάζεται η εταιρία.

Έχουμε 2 περιπτώσεις. Α) Τα λίτρα που χρειάζεται ακόμα η εταιρία είναι περισσότερα από αυτά που προσφέρει ο i αγελαδοτρόφος `if(N > a[i].second)`. Σε αυτή τη περίπτωση παίρνουμε όλα τα λίτρα που προσφέρει ο αγελαδοτρόφος Β) διαφορετικά παίρνουμε τα μόνο τα N λίτρα που χρειαζόμαστε.

```
#include<fstream>

#include<vector>

#include<algorithm>


using namespace std;


int main(){

    long int N,total=0;

    int M,t1,t2,i;

    vector< pair<int, int> > a;

    ifstream fin("milk.in");

    ofstream fout("milk.out");


    fin>>N>>M;


    for(i=0;i<M;i++){

        fin>>t1>>t2;

        a.push_back(make_pair(t1,t2));

    }

    sort(a.begin(),a.end());


    i=0;

    while(N>0){

        if(N>a[i].second){

            total=total+a[i].second*a[i].first;
```

```
        N=N-a[i].second;
    }
else{
    total=total+N*a[i].first;
    N=0;
}
i++;
}
fout<<total<<"\n";
return 0;
}
```