



IEEE Xtreme Countdown

Αναδρομή σε θέματα IEEE Xtreme

Μαρία Φωλίνα



Εισαγωγή στο Πρόβλημα

Mosaic Decoration II

IEEE Xtreme 14.0

Περιγραφή:

- Πρέπει να καλυφθεί ένας τοίχος διαστάσεων $W \times H$ χρησιμοποιώντας πλακάκια μωσαϊκού διαστάσεων $A \times B$.
- Δίνονται πλακάκια σε πακέτα των **10**, που κοστίζουν **M** το πακέτο.
- Αν τα πλακάκια δεν χωρούν ακριβώς στις άκρες του τοίχου, πρέπει να κοπούν. Κάθε κοπή κοστίζει **C** ανά ίντσα.
- Σκοπός: **ελάχιστο συνολικό κόστος** για να καλυφθεί ο τοίχος με πλακάκια και τις απαραίτητες κοπές

I/O και Παραδείγματα



→ **Είσοδος:** Έξι ακέραιοι σε μια γραμμή: **W, H, A, B, M, C.**

→ **Έξοδος:** Ένας ακέραιος, το ελάχιστο κόστος για την κάλυψη του τοίχου.

.....

Παράδειγμα:

- **Είσοδος:** **8 5 3 2 100 3**
 - Διαστάσεις τοίχου: **W = 8, H = 5**
 - Διαστάσεις πλακιδίων: **A = 3, B = 2**
 - Κόστος για 10 πλακάκια: **M = 100**
 - Κόστος για κάθε κοπή: **C = 3**
 - **Έξοδος:** **139**
-

Διαδικασία Λύσης



01

Υπολογισμός αριθμού πλακιδίων

Υπολογίζεται ο αριθμός πλακιδίων οριζόντια

$$(\text{rowtiles} = \text{ceil}(W/A))$$

και κάθετα

$$(\text{coltiles} = \text{ceil}(H/B))$$

Συνολικός αριθμός πλακιδίων: $\text{ntiles} = \text{rowtiles} * \text{coltiles}$

02

Υπολογισμός επιπλέον χώρου

Υπολογίζεται ο επιπλέον χώρος στις γραμμές

$$(\text{rowplus} = \text{rowtiles} * A - W)$$

και στις στήλες

$$(\text{colplus} = \text{coltiles} * B - H)$$

Αυτοί οι χώροι καθορίζουν πόσο χρειάζεται να κοπούν τα πλακάκια.

03

Υπολογισμός μήκους κοψίματος

Αν $\text{rowplus} <> \text{μηδέν}$, χρειάζεται κόψιμο κατά το ύψος του τοίχου, προσθέτοντας H στο συνολικό μήκος κοψίματος.

Αν $\text{colplus} <> \text{μηδέν}$, κοπή στο πλάτος, προσθέτοντας W.

Το $\text{συνολικό κόστος} = \text{μήκος κοπής} * C$

Υπολογισμός Κόστους



01

Κόστος Πλακιδίων

Καθώς τα πλακάκια πωλούνται σε πακέτα των 10, το κόστος είναι:

$$\text{tile_cost} = \text{ceil}(\text{ntiles} / 10) * M$$

02

Κόστος Κοπής

Το κόστος κοπής υπολογίζεται βάσει του μήκους κοπής που προσδιορίστηκε προηγουμένως:

$$\begin{aligned} \text{cutting_cost} = \\ \text{cutting_length} * C \end{aligned}$$

03

Συνολικό Κόστος

Το συνολικό κόστος είναι το άθροισμα του κόστους των πλακιδίων και του κόστους κοψίματος:

$$\text{price} = \text{tile_cost} + \text{cutting_cost}$$

Κώδικας σε Python



Βήμα 1: Ανάγνωση και επεξεργασία εισόδου

```
L = list(map(int, input().rstrip().split()))
```

```
W, H, A, B, M, C = L
```

Βήμα 2: Υπολογισμός αριθμού πλακιδίων

```
rowtiles = math.ceil(W / A)
```

```
coltiles = math.ceil(H / B)
```

```
ntiles = rowtiles * coltiles
```

Βήμα 3: Υπολογισμός επιπλέον χώρου και κοψιμάτων

```
rowplus = rowtiles * A - W
```

```
colplus = coltiles * B - H
```

```
cuttinginch = 0
```

Βήμα 4: Υπολογισμός κόστους κοψίματος

```
if rowplus != 0:
```

```
    cuttinginch += H
```

```
if colplus != 0:
```

```
    cuttinginch += W
```

Βήμα 5: Υπολογισμός συνολικού κόστους (πλακάκια + κοψίματα)

```
price = math.ceil(ntiles / 10) * M + cuttinginch * C
```

Βήμα 6: Εκτύπωση συνολικού κόστους

```
print(price)
```



Εισαγωγή στο Πρόβλημα (II)

Hotel Wiring IEEEExtreme 14.0

Περιγραφή:

- Σε ένα ξενοδοχείο με M ορόφους, κάθε όροφος έχει N δωμάτια.
- Λόγω ενός λάθους στην ηλεκτρική καλωδίωση, μερικά δωμάτια είναι συνδεδεμένα ανάποδα. Αυτό σημαίνει ότι ενώ ο διακόπτης του ορόφου είναι ενεργοποιημένος, τα δωμάτια αυτά δεν έχουν ρεύμα.
- Οι διαχειριστές του ξενοδοχείου θέλουν να μεγιστοποιήσουν τον αριθμό των δωματίων με ρεύμα, απενεργοποιώντας ακριβώς K διακόπτες ορόφων.

I/O και Παραδείγματα



- **Είσοδος:** M: αριθμός ορόφων, N: αριθμός δωματίων ανά όροφο, K: αριθμός διακοπών που θα κλείσουν.
- **Έξοδος:** Ο μέγιστος αριθμός δωματίων που λαμβάνουν ρεύμα μετά από το κλείσιμο των K διακοπών.

Παράδειγμα:

- **Είσοδος:**
 - Τεστ: **1**
 - Διαστάσεις: **2 2 1**
 - Σωστά δωμάτια: **2 0**
- **Έξοδος:** **4**



01

Ανάγνωση των Εισόδων

Για κάθε τεστ, εισάγουμε τα Μ, Ν και Κ καθώς και τον αριθμό των σωστά καλωδιωμένων δωματίων ανά όροφο.

02

Υπολογισμός των Δωματίων με Ρεύμα

Ξεκινώντας με όλους τους διακόπτες ανοιχτούς, προσθέτουμε τα σωστά καλωδιωμένα δωμάτια.

Για τους ορόφους όπου οι διακόπτες κλείνουν, υπολογίζουμε πόσα δωμάτια με λάθος καλωδίωση παίρνουν ρεύμα.

03

Βελτιστοποίηση

Τα δωμάτια που παίρνουν ρεύμα αυξάνονται αν κλείσουμε διακόπτες σε ορόφους με πολλά δωμάτια με λάθος καλωδίωση.

Χρησιμοποιούμε ταξινόμηση για να επιλέξουμε τους ορόφους που ωφελούνται περισσότερο από τα κλεισίματα.

Κώδικας σε Python



Είσοδος: Αριθμός τεστ

```
n = int(input())
```

```
for i in range(n):
```

```
    T = list(map(int, input().rstrip().split()))
```

Ανάγνωση M, N, K

```
J = []
```

```
sum = 0
```

```
for j in range(T[0]): # Διαδικασία για κάθε όροφο
```

```
    m = int(input())
```

```
    sum += m
```

```
    J.append(m)
```

```
J.sort() # Ταξινόμηση για επιλογή των K  
κλεισιμάτων
```

```
for k in range(T[2]):
```

```
    sum += T[1] - 2 * J[k]
```

```
print(sum)
```

Εκτύπωση του μέγιστου αριθμού δωματίων με
ρεύμα

$$\text{Net change} = (N - J[k]) - J[k] = N - 2 * J[k]$$

Όταν ο κεντρικός διακόπτης κλείνει:

- Τα λανθασμένα συνδεδεμένα δωμάτια αποκτούν ρεύμα, που είναι $N - J[k]$.
- Τα σωστά συνδεδεμένα δωμάτια χάνουν ρεύμα, που είναι $J[k]$.



Ευχαριστώ για την προσοχή σας

Επόμενο Workshop

20/10 IEEEEXTREME 16.0 17.0

