

# AtCoder Beginner Contest 106 解説

2018 年 8 月 18 日

E869120, square1001

For international readers: We usually don't have English editorial for ABCs, but this time only!

## 問題 A: Garden

答えから言うと、 $(a - 1)(b - 1)$  が答えとなります。これを導く方法は、主に 2 通りあります。

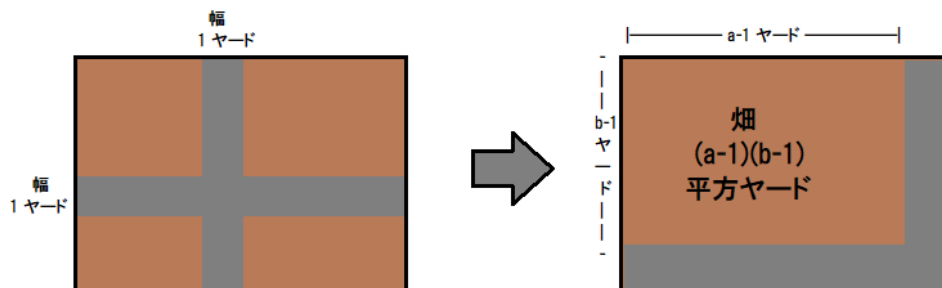
### 方法 1: 道路の面積から求める

道路の合計面積を求めるを考えます。

縦・横の道路両方重なっている部分が無ければ、 $1 \times a + 1 \times b = a + b$  が道路の面積の合計となります。しかし、実際は  $1 \times 1 = 1$  平方ヤード「縦方向・横方向両方の道路が交差している部分」が存在するので、道路となっている面積は  $a + b - 1$  平方ヤードとなります。答えは道路ではない面積なので、 $ab - a - b + 1 = (a - 1)(b - 1)$  となります。

### 方法 2: 図形をスライドさせる

以下の図のように、縦方向の道路を右端に、横方向の道路を下端に移動させるを考えます。



移動しても道路の幅が変わらないので、自明に求める面積は変わりません。移動した後の畑の面積は、縦  $a - 1$  ヤード横  $b - 1$  ヤードの長方形と同型となるので  $(a - 1)(b - 1)$  平方ヤードです。

### サンプルコード (C++)

<https://beta.atcoder.jp/contests/abc106/submissions/3024930>

## 問題 B: 105

この問題は、ABC 105 に出すつもりで作りましたが、結果的に ABC 106 に出されることになったようです。解法が大きく 2 つあるので、これを書きます。

### 解法 1: 全部の場合について計算

$C$  の約数の個数は、「 $C$  が 1 で割り切れるか?」「 $C$  が 2 で割り切れるか?」「 $C$  が 3 で割り切れるか?」…「 $C$  が  $C$  で割り切れるか?」という質問のうち答えが Yes である個数です。

「 $C$  が  $X$  で割り切れるか?」という質問は、「 $C$  を  $X$  で割った余りが 0 か?」という質問と同じなので、多くのプログラミング言語では「 $C \% X == 0$ 」かで判定ができます。For 文などを用いて  $X = 1, 2, 3, \dots, C$  までループすると、質問がそれぞれの  $X$  に対して実行できます。

それを、 $C = 1, 3, 5, 7, \dots, N$  (あるいは、 $C = 1, 3, 5, 7, \dots, N - 1$ ) に対して求めて、それぞれに対して約数が 8 個かどうかをチェックします。これにも For 文ループを使うことができます。このように、「2 重ループ」を書けばこの問題の答えを求めることができます。

### 解法 2: $N \leq 200$ までを利用する, ズルい解法

105 が最も小さい「約数が 8 個ある奇数」なので、200 までに「約数が 8 個ある奇数」が少ないことが想像できるでしょう。実は、約数が 8 個ある奇数は、105, 135, 165, 189, 195 の 5 つだけです。したがって、 $N \leq 200$  を仮定した場合、答えは次のように場合分けされます。

- $1 \leq N \leq 104$  のとき、答えは 0
- $105 \leq N \leq 134$  のとき、答えは 1
- $135 \leq N \leq 164$  のとき、答えは 2
- $165 \leq N \leq 188$  のとき、答えは 3
- $189 \leq N \leq 194$  のとき、答えは 4
- $195 \leq N \leq 200$  のとき、答えは 5

約数が 8 個ある奇数は、 $p, q, r$  を異なる素数として  $p^7$  の形、 $p^3q$  の形、 $pqr$  の形以外ありません。これが奇数であるとき、 $p, q, r$  はすべて奇数でなければなりません。200 以下に限定すると、 $pqr$  の形は  $3 \times 5 \times 7 = 105, 3 \times 5 \times 11 = 165, 3 \times 5 \times 13 = 195$  の 3 個あり、 $p^3q$  の形は  $3^3 \times 5 = 135, 3^3 \times 7 = 189$  の 2 個あります。

ソースコード (C++)

解法 1: <https://beta.atcoder.jp/contests/abc106/submissions/3024975>

解法 2: <https://beta.atcoder.jp/contests/abc106/submissions/3024982>

## 問題 C: To Infinity

まず、5000 兆回の操作で 1 文字の文字列がどのように変わるかを考えます。

- "1" → "1" → "1" → "1" → ... → "1"
- "2" → "22" → "2222" → "22222222" → ... → (2 が  $2^{5000}$  兆 文字続く文字列)
- "3" → "333" → "333333333" → ... → (3 が  $3^{5000}$  兆 文字続く文字列)
- "4" → "4444" → "4444444444444444" → (4 が  $4^{5000}$  兆 文字続く文字列)
- 5~9 に関しても同じように、5000 兆回操作を繰り返すと  $K$  文字より明らかに大きくなる

つまり、5000 兆回操作を行った後の文字列は以下のように考えられます。

- "1" はそのまま、変わらない
- "2" 以上の文字  $A$  は全て、 $A$  だけから成る無限長の文字列に変化する

よって、以下のことが言えます。

- 文字列  $S$  において、1 文字目から  $K$  文字目まで全て '1' であれば、答えは '1'
- そうでなければ、答えは  $S$  中に初めて出現する '2' 以上の文字

実装に関しては、For 文で最初の文字から順に  $K$  文字目まで探し、"1" 以外の文字が出てきたときに打ち切る、とすれば簡単にできます。300 bytes 以内の簡単な実装も容易です。

### サンプルコード (C++)

<https://beta.atcoder.jp/contests/abc106/submissions/3024955>

## 問題 D: AtCoder Express 2

各列車の走る区間は  $(L_j, R_j)$  であり、クエリは  $(p_i, q_i)$  であるように、それぞれについて「区間の左端」「区間の右端」の 2 つの要素しかないので 2 次元座標として見做すことを考えましょう。

さて、 $x_{l,r}$  を 区間  $[l,r]$  を走る列車の個数、とします。クエリ  $(p,q)$  に対して、求めたい値は  $x_{p,p} + x_{p,p+1} + \dots + x_{p,q} + x_{p+1,p} + \dots + x_{q,q}$  となります。つまり、二次元区間に落とし込むと、求めたいものは、例えば囲まれた区間の合計となります。



これを高速に求めるためには、累積和を使うと上手くいきます。この問題は  $N \leq 500$  と小さいため、累積和を全て記録しても実は間に合います。

$c_{i,j} = x_{i,1} + x_{i,2} + \dots + x_{i,j}$  とします。そうすると、 $x_{i,l} + x_{i,l+1} + \dots + x_{i,r} = c_{i,r} - c_{i,l-1}$  として表されるので、各クエリにつき  $O(N)$  で解けます。 $Q \leq 100000, N \leq 500$  より、実行時間制限の 3 秒には余裕で間に合うはずで、計算量は  $O(QN + N^2)$  です。

ちなみに、二次元累積和を使うとさらに高速化できます。それを用いると  $O(N^2 + Q)$  が達成できますが、本問題ではここまでは要求されません。

### サンプルコード

[1] 解法 - <https://beta.atcoder.jp/contests/abc106/submissions/3024990>

[2] 二次元累積和を用いる解法 -

<https://beta.atcoder.jp/contests/abc106/submissions/3024983>

## A – Garden

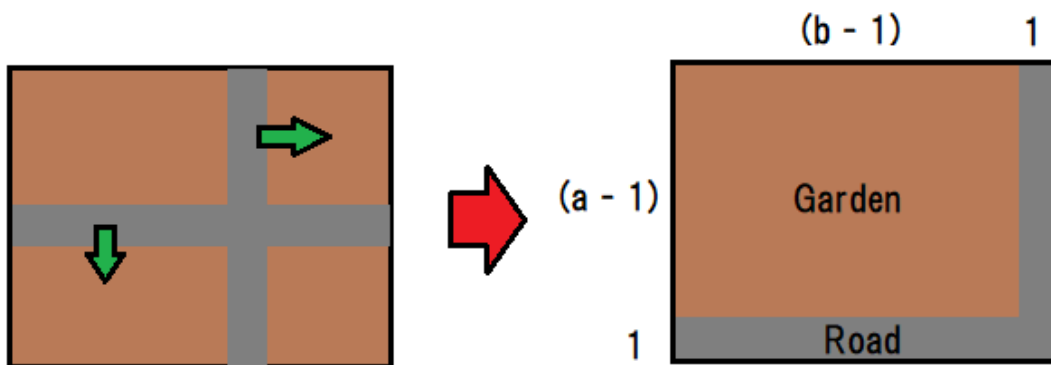
First of all, the answer of this problem is  $(a - 1)(b - 1)$ . Here is why this is correct:

[Way 1]

Let's think about the area which covered with road. If there was no area which is covered with both road which are horizontal and vertical, the area should be  $a + b$ . However,  $1 * 1 = 1$  square yard is covered with both two roads. So, the answer of this problem is  $ab - (a + b - 1) = (a - 1)(b - 1)$ .

[Way 2]

Let's think about moving horizontal road to right, and vertical road to left. Obviously, the garden will become a rectangle of  $(a - 1) * (b - 1)$ , so the area is obviously  $(a - 1) * (b - 1)$ .



### Sample Code

- <https://beta.atcoder.jp/contests/abc106/submissions/3024930>

## Problem B: 105

This problem has a background that we made this problem to publish in AtCoder Beginner Contest 105, but it was published in AtCoder Beginner Contest 106.

### Solution 1: Compute for all possible cases

First, let's think about counting the number of divisors of  $C$ . We can divide  $C$  into  $1, 2, 3, \dots, C$ , and check the remainder respectively. If the remainder of dividing into  $X$  is zero, it means  $X$  is a divisor of  $C$ . It is convenient if we use loop (for example, for-statement).

Now, we can check for  $C = 1, 3, 5, 7, 9, \dots, N$  (or,  $C = 1, 3, 5, 7, 9, \dots, N - 1$  if  $N$  is even). We can also use loop like for-statement. It means, one way to implement the solution is, using "double-loop".

### Solution 2: Tricky solution using $N \leq 200$

Using "105 is the least odd integer that there are exactly 8 divisors", we can imply that there are only few odd integers below 200, that there are exactly 8 divisors.

Actually, the only such numbers below 200, are 105, 135, 165, 189, 195. Assuming  $N \leq 200$ , the answer can be divided as follows:

- If  $1 \leq N \leq 104$ , the answer is 0
- If  $105 \leq N \leq 134$ , the answer is 1
- If  $135 \leq N \leq 164$ , the answer is 2
- If  $165 \leq N \leq 188$ , the answer is 3
- If  $189 \leq N \leq 194$ , the answer is 4
- If  $195 \leq N \leq 200$ , the answer is 5

The numbers that there are exactly 8 divisors, can be represented with three different prime numbers  $p, q, r$ . There are only three patterns:  $p^7, p^3q, pqr$ . There are no  $p^7$  pattern below 200. There are two  $p^3q$  patterns,  $3^3 \times 5 = 135$  and  $3^3 \times 7 = 189$ . There are three  $pqr$  patterns,  $3 \times 5 \times 7 = 105, 3 \times 5 \times 11 = 165, 3 \times 5 \times 13 = 195$ .

### Source Code (C++)

Solution 1: <https://beta.atcoder.jp/contests/abc106/submissions/3024975>

Solution 2: <https://beta.atcoder.jp/contests/abc106/submissions/3024982>

## C – To Infinity

Let's think about the state of after  $5 * 10^{15}$  days, for **one character**.

- '1' -> '1' -> ... -> '1'
- '2' -> '22' -> '2222' -> '22222222' -> ... ( $2^{5*10^{15}}$  characters)
- '3' -> '333' -> '333333333' -> ... ( $3^{5*10^{15}}$  characters). And so on.
- For '2', '3', ..., '9', it will become more than  $K$  characters obviously.

So, you can assume that character '2' – '9' will be infinite length, because it's more than  $K$  characters. You can say as follows:

- If all characters between 1<sup>st</sup> to  $K$ -th character is '1', the answer will be '1'.
- Otherwise, the answer will be the character of leftmost index which is not '1'.  
For example, if  $K = 10$  and  $S = "113123"$ , the answer will be 3.

Implementation is easy. You can write with less than 300 bytes.

### Sample Code

<https://beta.atcoder.jp/contests/abc106/submissions/3024955>

## Problem D: AtCoder Express 2

Let's think about placing trains into 2-dimensional plane. Think about placing train  $i$  in coordinate  $(L_i, R_i)$ . Then, for each query, since we have to find the number of  $i$  that  $L_i \geq p$  and  $R_i \leq q$ , it means we should count the number of "trains which is right-lower than  $(p, q)$ ".

We can use dynamic programming (also known as "two-dimensional cumulative sum technique"). Since  $N$  is no more than 500, the coordinate is positive and less or equal to 500.

Now, let  $C_{i,j}$  the number of trains in  $(i, j)$ . The answer  $a_{i,j}$  when  $(p, q) = (i, j)$  is, if you think carefully... It will be  $a_{i,j} = C_{i,j} + a_{i,j-1} + a_{i+1,j} - a_{i+1,j-1}$ .

Then, if we calculate from the larger x-coordinate and smaller y-coordinate, you can calculate all  $a$  where  $1 \leq x \leq N, 1 \leq y \leq N$ , in  $O(N^2)$  time.

Now, the answer of query is stored in  $a_{p,q}$ . It means you use only constant time for each query. The total time complexity is  $O(N^2 + M + Q)$ .

Source Code (C++):

<https://beta.atcoder.jp/contests/abc106/submissions/3024983>