Finding Correlated Pairs

Martin Aumüller

October 26, 2020

1 Input

The input is a list of 256-bit vectors, one per line, from stdin. Each vector is divided into four 64-bit parts; each part is store as a signed 64-bit integer. For example, the line

```
2948679682100370091 -6730236453359443949 9154238340659291137 2505149300205180166
```

gives the bit vector (here I am breaking into 4 lines for readability; the bits give a single total vector):

```
00101000111010111101000010001100010100000000000100011010101011
10100010100110010110010110011010100111110010000000100000000010011
01111111000010100110001101000101000001000100001000100100000000001
00100010110001000001010000000100000001000000110001010101000000110
```

The very first line from stdin is an integer that represents the number of vectors that follow.

1.1 How Vectors Were Generated

Each bit of every vector is a `1` with probability 1/3, and a `0` with probability 2/3. These choices are made independently.

The correlated vector (we'll call it v_c for simplicity) is *Pearson-correlated* with another vector v. For each digit in v_c, with probability 7/8 we set it equal to the corresponding digit in v. With probability 1/8, the digit is set randomly—`1` with probability 1/3, and `0` with probability 2/3. The Pearson-correlated vectors are much more likely to have a large number of `1`s in common than two random vectors.

2 Correlation

We define the *similarity* of two vectors to be the number of `1`s they have in common. We also call them *correlated*.

We guarantee that in each input instance, there is a single pair with similarity at least 70.

As an example, the following two vectors have similarity 93. If we find these two inputs in a file, we can safely output them as the correct answer.

```
-4545410128706124432 423485083363186329 28290711320299080 -1687296419800652167
-4256053852647568032 428569225130022555 28290711253190152 -1687296419734591911
```

3 Expected Output

The expected output on CodeJudge is the two indices of the correlated vectors on one line, with the lower index first.