IEEEXtreme 10.0 > Binary Quilts

# Binary Quilts

locked

by *IEEEXtreme*

Problem | Submissions | Leaderboard | Discussions | Editorial

One of the first considerations when trying to solve this problem is how one should represent a matrix. One of the most compact and convenient ways would be to store the 5x5 binary matrix as a 25 bit number, which fits easily into our 32-bit signed integers. The maximum value, which corresponds to the number of matrices would be $2^{25}$ or about 33 million.

A breadth-first search could be used to determine the length of the shortest sequence of operations. Since the matrices are represented as integers between 0 and $2^{25}$-1, one could use an array of values to indicate whether a matrix has been visited by the search, and if so, what the distance to the matrix was.

The problem with this approach is that the large number of transitions from each state. The total number of operations possible on a matrix is 115:

- 25 negate operations on 1x1 squares (note that the flip operations do nothing for these regions)
- 48 operations on 2x2 squares (16 flip vertical, 16 flip horizontal, and 16 negate)
- 27 operations on 3x3 squares (3 operations and 9 3x3 squares)
- 12 operations on 4x4 squares (3 operations and 4 4x4 squares)
- 3 operations on 5x5 squares (3 operations and 1 5x5 squares)

An upper bound on the number of possible transitions would be 115 * $2^{25}$ or approximately 3.9 billion. This is too large a number for a program to be able to consider within the time constraints.

However, one could take a meet in the middle approach, aka bidirectional search, to greatly reduce the number of transitions considered. In this approach, we use two simultaneous searches, one started from the start matrix, and one from the end matrix. When the frontiers of these searches overlap, we can determine the shortest path.

Note that in order to pass all of the testcases within the time constraints, you will likely need to use bit operations to make the transitions. All of the negate operations can be written with a single bitwise xor statement. For example, to negate the last bit in the integer in Java or C++, this would be written as:

```
board ^ 1
```

## Statistics

**Difficulty: Hard**
**Publish Date: Jul 11 2016**

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Scoring | Environment | FAQ | Terms Of Service | Privacy Policy