

code jam

print "hello, world!"

Qualification Round 2017

[A. Oversized Pancake Flipper](#)[B. Tidy Numbers](#)[C. Bathroom Stalls](#)[D. Fashion Show](#)

Contest Analysis

[Ask a question](#)[View my submissions](#)

Submissions

Oversized Pancake Flipper

5pt	Correct 19628/23634 users correct (83%)
10pt	Correct 17799/19074 users correct (93%)

Tidy Numbers

5pt	Correct 24253/26071 users correct (93%)
15pt	Correct 17756/22162 users correct (80%)

Bathroom Stalls

5pt	Not attempted 13983/16043 users correct (87%)
10pt	Not attempted 10822/13227 users correct (82%)
15pt	Not attempted 5954/8864 users correct (67%)

Fashion Show

10pt	Not attempted 996/2522 users correct (39%)
25pt	Not attempted 591/843 users correct (70%)

Top Scores

FatalEagle	100
ACMonster	100
y0105w49	100
johngs	100
HellKitsune123	100
SergeyRogulenko	100
spnautilus	100
BudAlNik	100
mjy0724	100
pwild	100

Practice Mode Rank: 13551 Score: 35

asprazz | [Contest scoreboard](#) | [Sign out](#)

Contest Analysis

[Overview](#) | [Problem A](#) | [Problem B](#) | [Problem C](#) | Problem D

Fashion Show: Analysis

What's really going on here?

The somewhat strange scenario presented in this problem disguises a classic type of chess problem in which pieces must be placed so that they do not attack each other. The [eight queens puzzle](#) is one well-known example. Our problem is more complicated, and it involves three types of piece. Let's restate the rules about how the models can interact in more chess-like terms:

- The + models are bishops. Two bishops may not occupy the same diagonal.
- The x models are rooks. Two rooks may not occupy the same row or column.
- The o pieces are queens. Two queens may not occupy the same row, column, or diagonal. Moreover, a queen and a bishop may not occupy the same diagonal; a queen and a rook may not occupy the same row or column.

Observe that our problem does not accord with typical chess rules about "attacks". For example, in our problem, it is fine for a rook and a bishop to share the same row or column. Also, unlike in chess, a piece between two pieces does not prevent them from "attacking" each other. For example, we do not allow two bishops to share the same diagonal even if there is a rook between them.

Decomposing the problem

How will we deal with this variety of pieces? The critical insight is that the rook and bishop parts of the problem are independent; we can solve them separately, placing as many new pieces in each subproblem as possible, and then merge the answers together. A queen is just a rook plus a bishop; we can add each pre-placed queen into the rook subproblems as a rook and into the bishop subproblem as a bishop. Then, once the subproblems are solved, we can turn any cell that is occupied in both subproblem solutions back into a queen.

This strategy is guaranteed not to violate any rules. A rook subproblem solution will never have two rooks in the same row or column, and a bishop subproblem solution will never have two bishops on the same diagonal. Merging the two subproblems' solutions may generate new queens, but it is impossible for them to violate any rules, since that would imply a rule violation in one of the subproblems. For example, we do not need to worry that we will end up with a queen and a bishop on the same diagonal, since that would only be possible if our bishop solution had two bishops on the same diagonal.

Moreover, as long as we place as many rooks as possible in the rook subproblem, and as many bishops as possible in the bishop subproblem, we are guaranteed the maximum possible amount of style points for our test case. Since queens are worth 2 points, merging a rook and a bishop into a queen has no effect on our score.

Let's walk through an example. This case:

```
+..
+.0
x..
```

can be decomposed into rook and bishop problems:

```
... +..
..x +.+
x.. ...
```

which can be solved independently:

```
.x. +..
..x +.+
x.. +..
```

and then merged back together. Note that we have replaced the former x model from the lower left corner with an o model.

```
+x.
+.0
o..
```

Solving the subproblems

Now all we need are strategies for the subproblems themselves. The rook problem is straightforward. Each rook removes exactly one row and column from further consideration, so any greedy placement strategy that does not violate the rules will place exactly **N** rooks.

The bishop subproblem is more challenging. We can approach it differently in the Small and Large datasets.

Bishops: Small dataset

In the Small dataset, any pre-placed pieces are all in the top row. Observe that bishops in the same row or column cannot possibly "threaten" each other, and so we can safely add

a bishop to any top row cell that does not already have one. So, if we can come up with a general solution pattern in which the top row is always completely filled with bishops, then we can solve any Small test case, because we can safely turn any pre-placed arrangement into a row packed with bishops.

Once the top row is filled with bishops, where else on the board should we put them? The bottom row is farthest away from the constraints imposed by the top row, and we can try putting a bishop in every bottom-row cell except for the cells on either end (which are "threatened" by the bishops at the two ends of the top row). These bishops do not threaten each other or any top row bishops, so this arrangement is valid, and we have a total of $2N-2$ bishops. No additional bishops can be added after that, though. Have we really placed as many as possible?

At this point, we can experiment and convince ourselves that this solution is probably optimal. We can also take a leap of faith; for a Small dataset in a Qualification Round, there is little incentive not to submit the solution and see if it is correct. Or, we can come up with a proof. An N by N board has $4N-2$ different diagonals. Moreover, the parallel diagonals of length 1 in opposite corner cells can never both be used simultaneously, so there are really only $4N-4$ simultaneously usable diagonals. Since placing a bishop uses up two diagonals, $2N-2$ is an upper bound on the number of bishops we can place. So, our method is optimal!

We must still take care, though, to handle a pre-placed rook/queen correctly if one is present, and to merge the rook and bishop solutions appropriately, creating queens when necessary. We must also be careful with the 1 by 1 board, which has no bottom row distinct from its top row.

It is possible to come up with the same construction without realizing that the problem can be decomposed into rooks and bishops; it is just more difficult to justify the optimality of the construction in that case!

Bishops: Large dataset

One helpful observation is that, just as in a chess game, the "white cell" bishops (in our problem, cells for which $R_i + C_i$ is even) are completely independent of the "black cell" bishops (cells for which that sum is odd). So we can consider these as sub-sub-problems.

You can use a [bipartite matching algorithm](#) to place the bishops optimally. There is, however, a greedy strategy; unlike in the rook subproblem, however, not just any greedy strategy will work!

Let's consider an 8 x 8 board with no pre-placed bishops. We'll look at just the "black" cells of the board, and tilt the board 45 degrees clockwise. (Here, `.`s represent black cells. The `@`s do not represent cells — they are just there to orient the image.)

```
@@@. .@@@
@@. . . .@@
@. . . . .@
. . . . .
@. . . . .@
@@. . . .@@
@@@. .@@@
```

This new board has an important property: any row is a subset of all rows with more black cells than it. For example, the four black cells in the second row are also present in every row with at least four black cells. This property holds regardless of the value of N , or whether we look at "white" or "black" cells. It even holds as we add bishops! Adding a bishop wipes out one entire row and one entire column — notice that we have made this more like the rook problem — and since the remaining rows have all lost the same column, the aforementioned property is unchanged.

The property suggests a greedy strategy: first, sort the rows by the number of available cells. Then, pick a "smallest" row (one with a minimal number of available cells), place a bishop in any column in that row, and wipe out that row and column. This strategy is guaranteed to place an optimally large number of bishops. Suppose that we choose a column C in the smallest row R , and another column C' in some other row R' , such that C' is also in R . Then C must also be in R' , since all columns in the smallest row are in every other row. We can therefore swap them over, and it is equally valid to choose C' in row R and C in row R' .

One tempting (but incorrect) greedy strategy is to go left to right, top to bottom, and greedily place bishops in all legal places. Here is an example of a board for which that fails.

```
.@@@
...@
...@
..@@
.@@@
```

The incorrect greedy strategy will only place three bishops, as follows:

```
+@@@
+. .@
..+@
..@@
.@@@
```

whereas the correct strategy will place four (here is one optimal placement):

```
.@@@
...+
..+@
.+@@
+@@@
```

Notice that although we can always place **N** rooks, the number of bishops we are able to place depends on the pre-placed bishops. This explains why different test cases with the same value of **N** might have different maximum scores.

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform