Actually, I didn't succeed in finding the solution to this question.

When I saw this question, I thought it was a question of map coloring, and tried to solve it with the method of map coloring. However, I didn't succeed in it.

Now, I have some ideas to solve this problem, but I haven't found a suitable way to implement it in code.

My ideas are:

First, put all the beads of different colors into the grid in random.

Second, define a variable to record the minimum penalty and define a matrix to record the distribution of beads when the minimum penalty is taken.

Third, construct a recursive function. In each recursion, two beads in the grid are exchanged. And after each exchange, the total penalty is recalculated. If the total penalty is smaller than the minimum penalty, the total penalty and the distribution of beads will be recorded.

Finally, when the recursive function is finished, the distribution of beads will be the final result.

Result:

1. L=5 with 12 red beads (R) and 13 blue beads (B)

```
BRBRB
RBRBR
BRBRB
RBRBR
BRBRB
```