Algorithm Design Term Project

Chanatip Kwankeua 6215452
ZhengLi Zhang 6218016

Presentation Agenda

1. Problem Statement
2. Brute Force Idea
3. Solution

Problem Statement

BISHOPS

Determine the maximum number of bishops that can be placed on a chessboard in such a way that no two bishops threaten each other.

Input: Each line represent the size of the board. (Board size is $1 < N < 10^{100}$)

Expected output: The maximum number of bishops corresponding to the input in order.

Example:

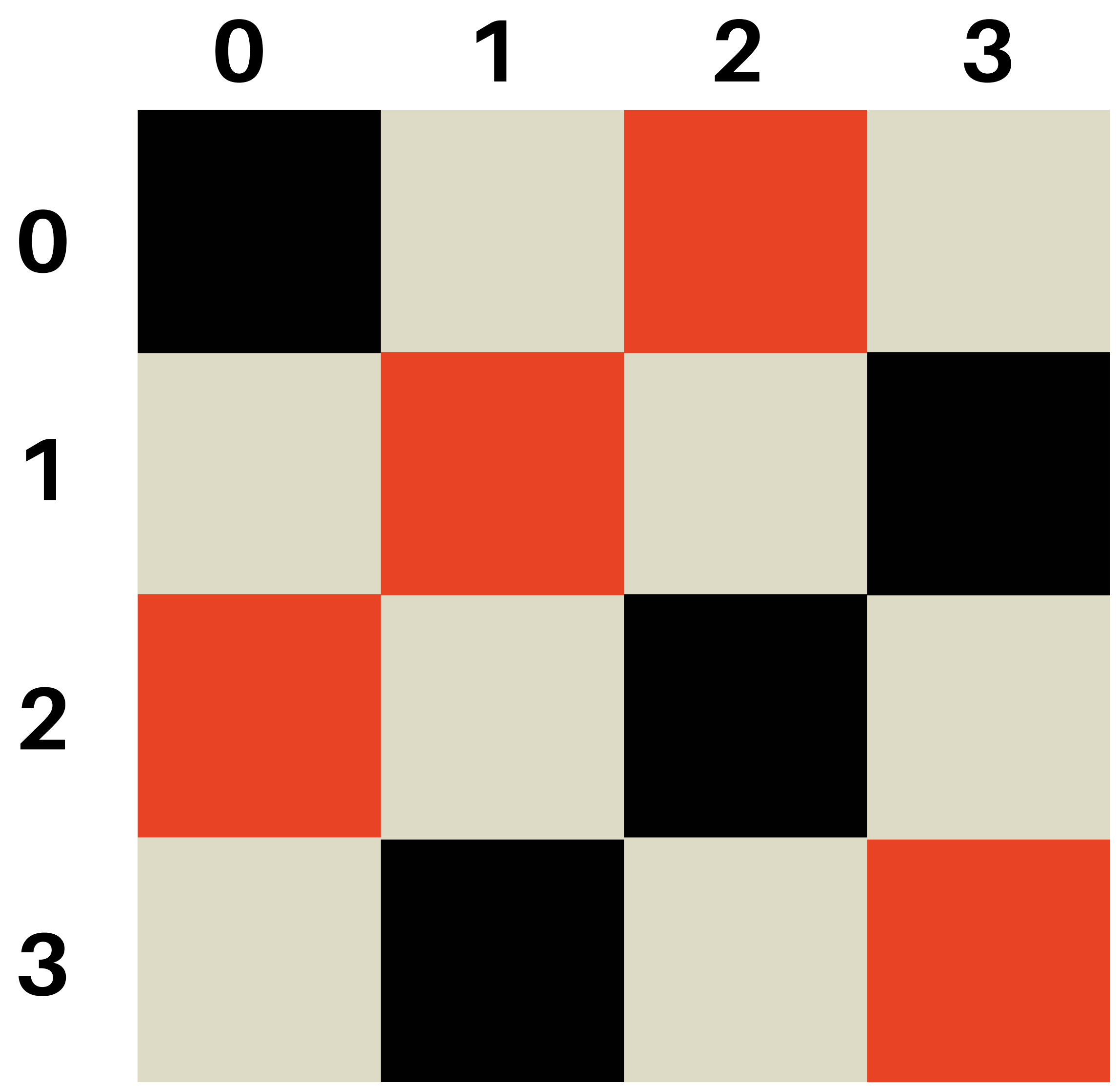
```
Input:
2
3

Output:
2
4
```

Brute Force

Idea

1.Assign each side of the board a variable to iterate through.

For example: The column will be X starting from 0 to n-1 (size-1) and the row will be y starting from 0 to n-1 (size-1)

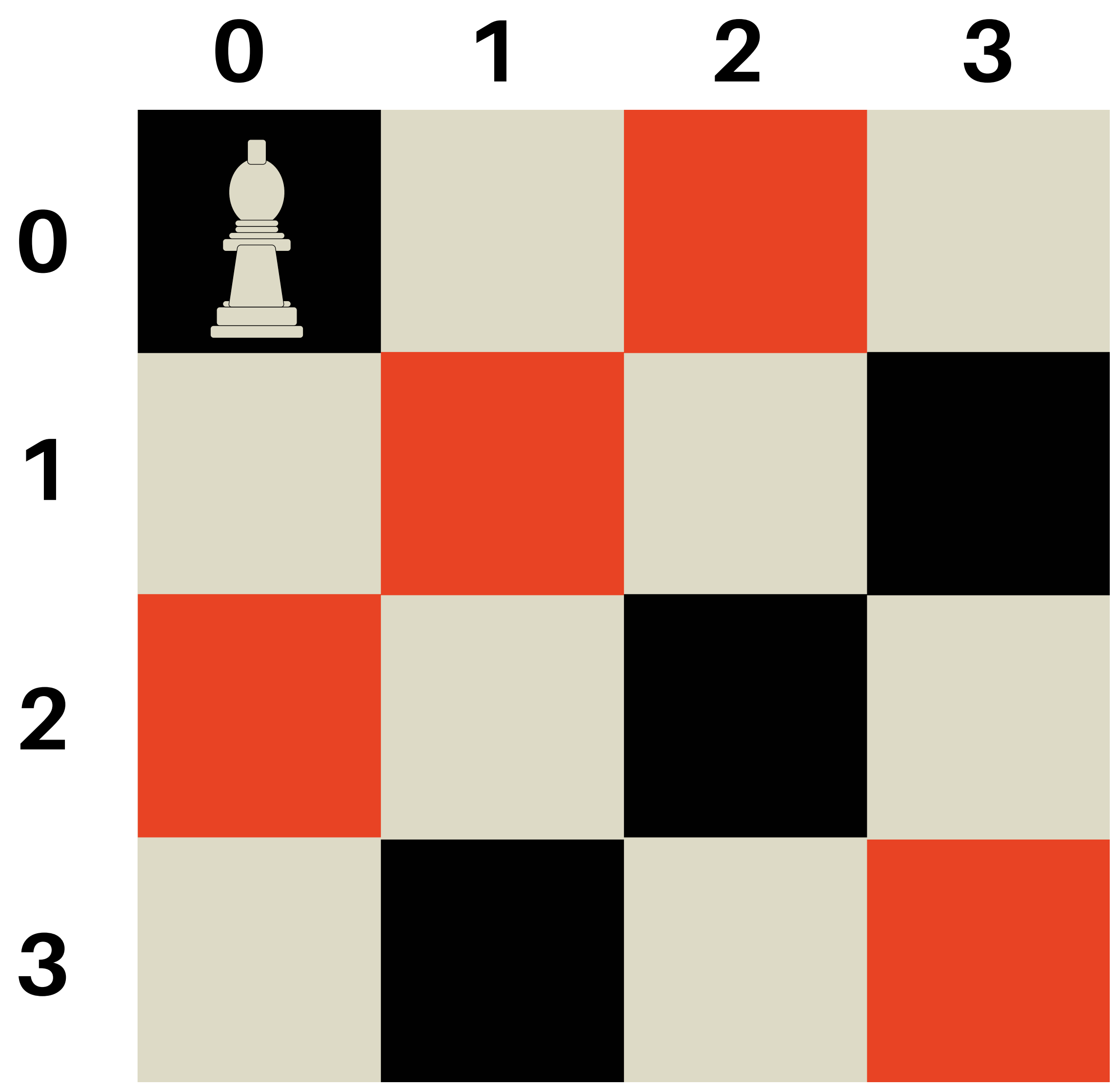


Brute Force

Idea

2. Iterate through I and J one by one and choose between placing the bishop or not placing the bishop. If the bishop is placed than a counter goes up by one.

Note: A decision tree can be made from the choice to place the bishop (represent by 1) and not to place the bishop (represent by 0)

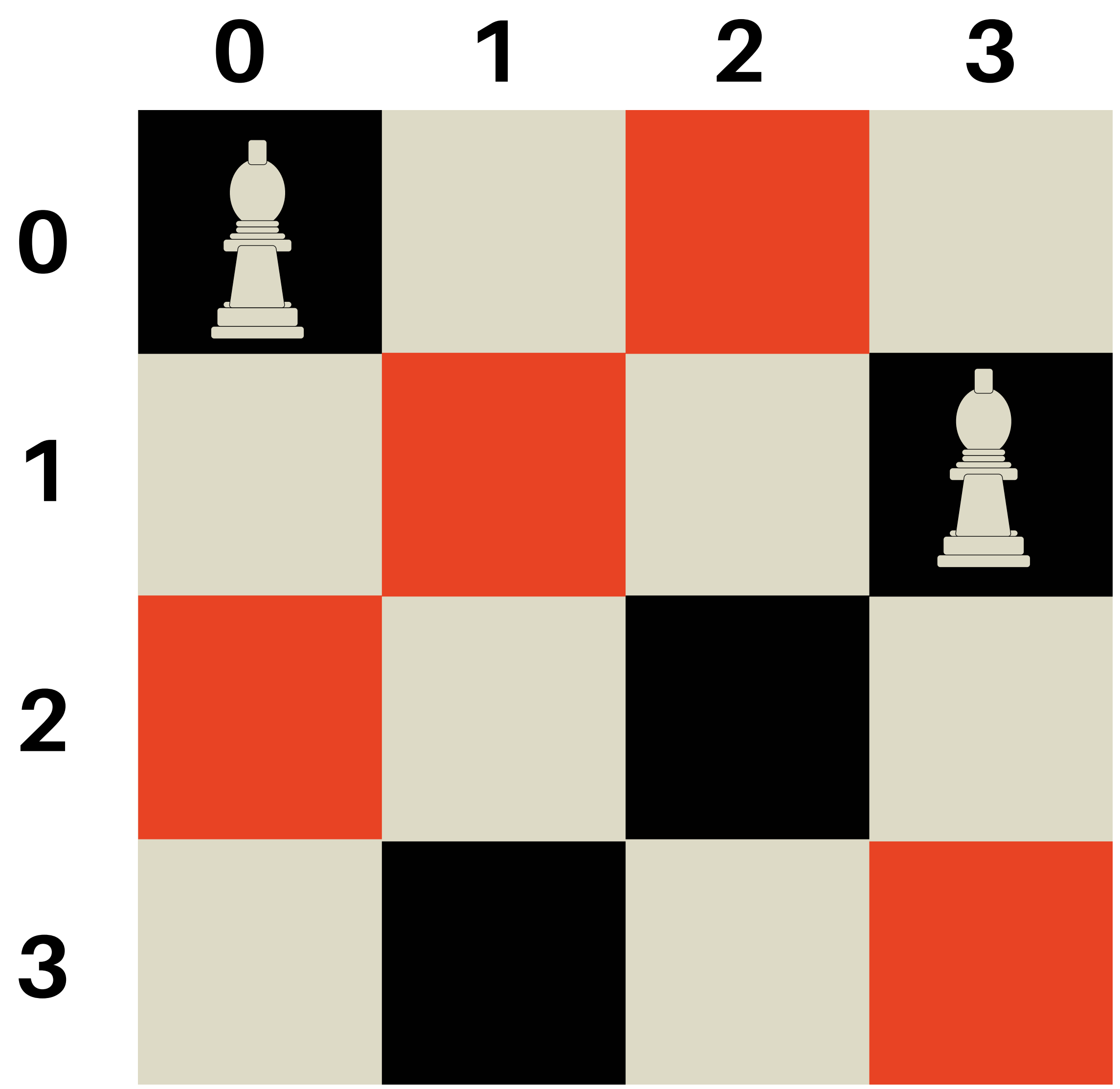


Brute Force

Idea

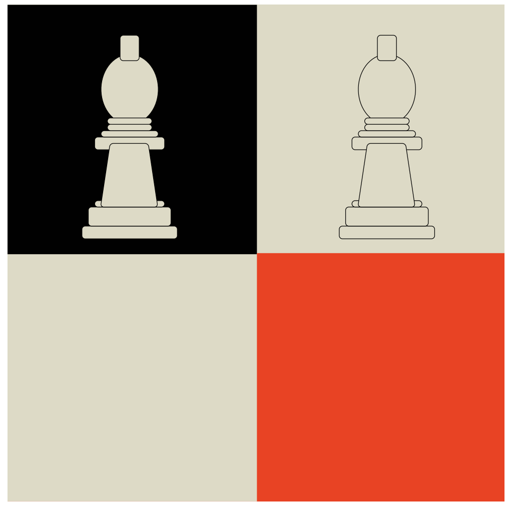
3. Allow the algorithm to do recursion to see all the possible solutions and compare the results to find the maximum number of bishops placed for each cases.

Note: This method will be slow if the board size gets bigger because N can be as big as 10^100.



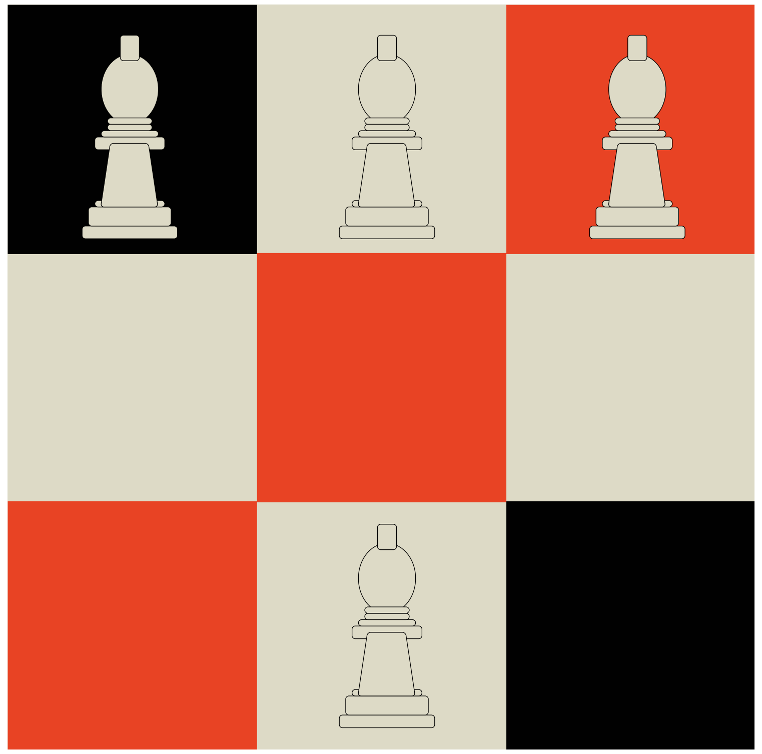
Solution

By illustrating the simple questions, we can see the patterns within the solution.



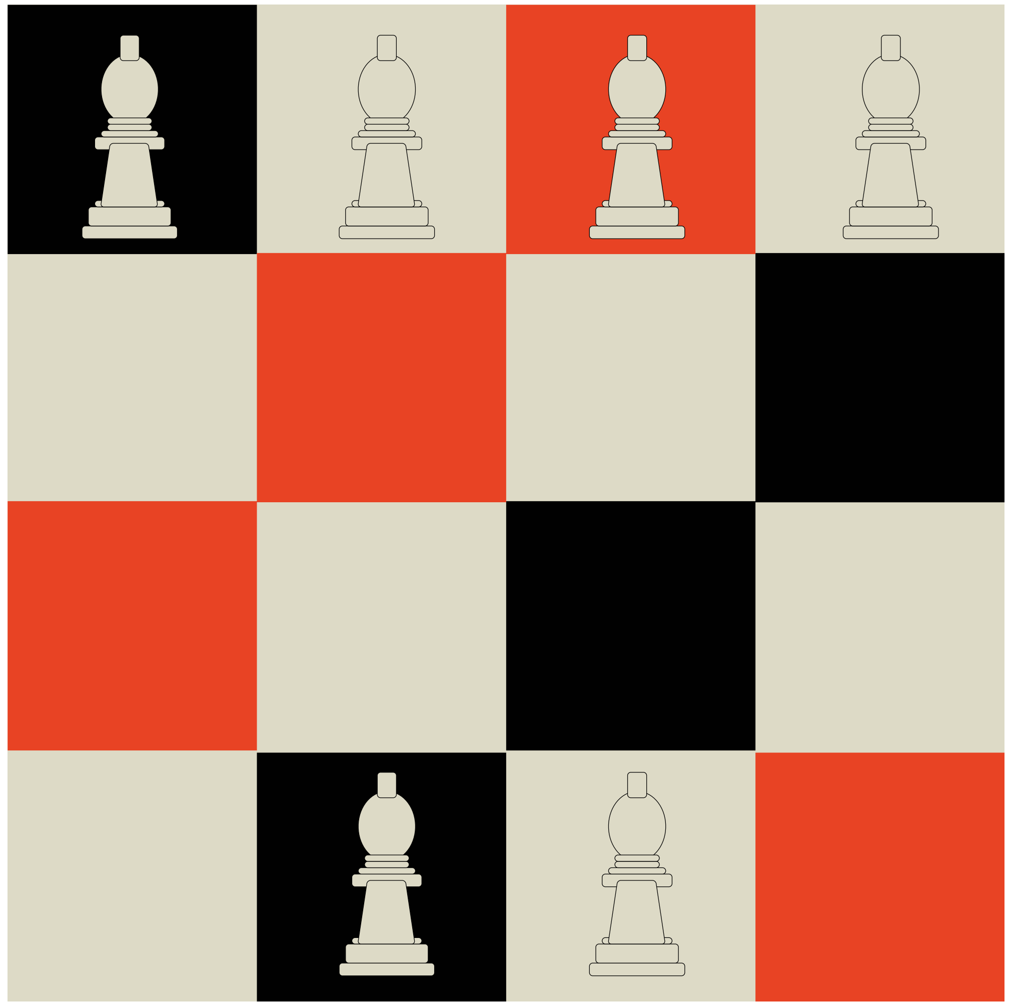
N = 2

S = 2



N = 3

S = 4



N = 4

S = 6

The correlation between N and the result is $S = N * 2 - 2$

The Code and Result

```
l = []
while True:
    try:
        n = int(input())
        l.append(n)
    except:
        break
for i in l:
    if i > 1:
        print(i*2-2)
    else:
        print(1)
```

28454602	2021-09-22 17:57:49	Guy	Bishops	**accepted** edit ideone it	0.03	9.0M	PYTHON3

Reference: https://www.spoj.com/problems/BISHOPS/