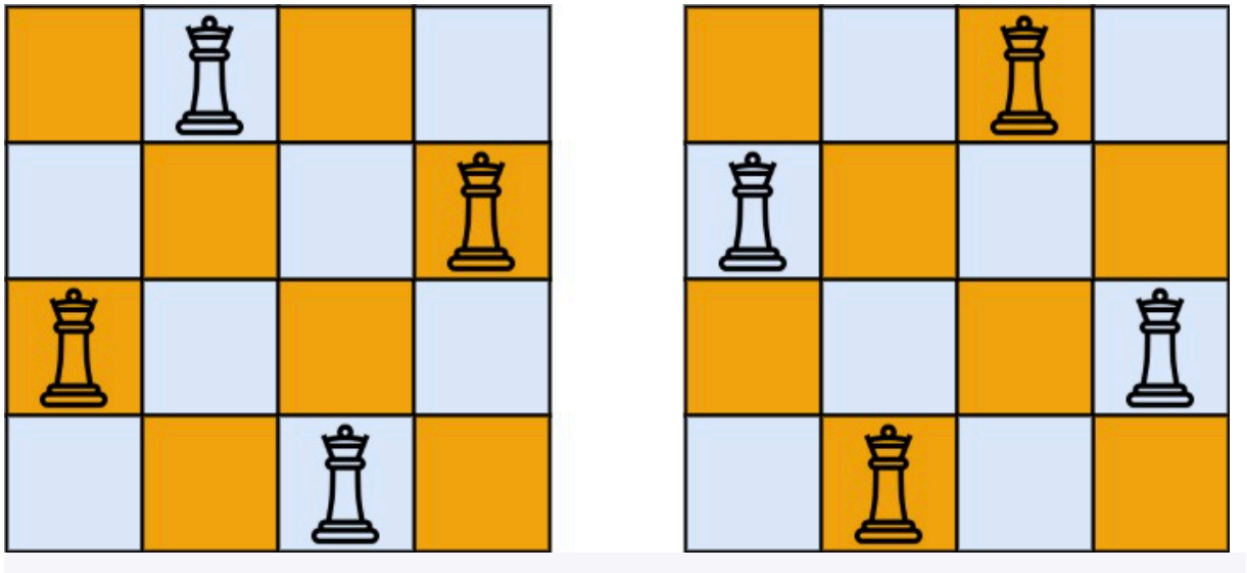
Alexandra Holguin anh2150

Project Proposal - N-Queens

The project I plan to implement is a common problem known as N-Queens. The idea behind the project is simple: to calculate the number of ways that *n* Queens can be arranged on an *n*x*n* chessboard in ways that none of queens are threatening another queen.

Another way to think about it the number of ways to rearrange the queens in a way such that there is only one queen per row, per column and per diagonal.

For example, for $n = 4$, there are two possible solutions:



The problem of n-queens fairly simple for small values of *n*. But the calculations quickly balloon into astronomical proportions as the value of *n* increases. As such, it should be a very good example of leveraging the cores and threads of modern processors and solve the problem in parallel.

Before implementing the problem, I would guess that solving the problem would approximately cut the amount of time by the number of parallel threads.

Another challenge with this problem is to minimize the total amount of RAM utilized. Some quick online research suggests that there are ways to use bit wise operations to use small numeric representations to minimize RAM usage. But, it remains to be seen how the problems translates to a purely functional language like Haskell.