

Theory Problems 8

Algorithm Design

Problem 1: Big multiplication

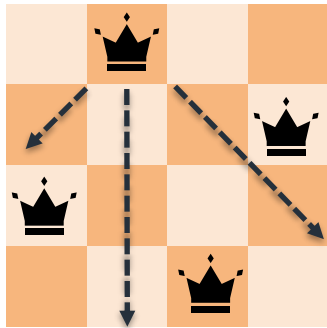
In lecture 8, we saw that multiplication could be implemented using divide and conquer by dividing up the digits. Here we will use binary instead of denary

$$a \cdot b = (a_0 + a_1 2^{n/2}) \cdot (b_0 + b_1 2^{n/2}) = a_0 b_0 + a_0 b_1 2^{n/2} + a_1 b_0 2^{n/2} + a_1 b_1 2^n$$

1. Show that we can evaluate the formula on the right with just three multiplications (and some additions/subtractions)
2. Write an algorithm to multiply two arbitrary-length integers using divide-and-conquer
3. What is the time complexity of your algorithm?

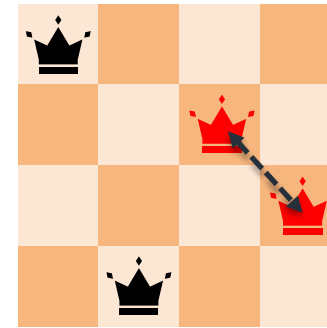
Problem 2: n-Queens

Problem: Place n queens on an $n \times n$ chessboard so that no queen attacks another.



Valid

$$n = 4$$



Invalid

Write a backtracking algorithm in Java to find all solutions of the n-queens problem for a given n .

Problem 3: Improved n-Queens

1. Improve your n-Queens algorithm by using a pruning strategy.
2. You are now only required to produce one solution. Implement a heuristic to speed up your algorithm.