Find two values of n for which the NQueens problem has no solutions. Explain why in detail, in your discussion use some diagrams.

When initializing NQueens, we specify an n number of queens we would like on a n by n sized board. When we specify n to be either 2 or 3, we find that there are no spaces left for the final queen to be placed. In other words, there are no solutions at n = 2, and n = 3.

As shown in Diagram 1, a second queen cannot be placed onto a 2x2 board, once the first queen is placed. There are no open white spaces for queen 2 to be placed and be safe from queen 1's attacks.

Within Diagram 2.1, we see that there are two open spaces available for a queen to occupy. However, once we put down a second queen to occupy one of these spaces (Diagrams 2.2, 2.3,) we find that there are no white spaces left for the final queen to occupy. We do, however, see a lot of overlapping spaces (depicted in purple) such that both queens can occupy at any point in time.

Diagram 1

Possible movement depicted by queen 1 occupation.

Queen 1 placement	Queen 1 row movement
Queen 1 column movement	Queen 1 diagonal movement

Diagram 2.1

Possible movement depicted by queen 1 occupying the first column of row 1.

Queen 1 placement	Queen 1 row movement	Queen 1 row movement
Queen 1 column movement	Queen 1 diagonal movement	Open for queen occupation
Queen 1 column movement	Open for queen occupation	Queen 1 diagonal movement

Diagram 2.2
Possible movement depicted if the bottom middle space was occupied by queen 2.

Queen 1 placement	Occupied by Queen 1 and 2's movement	Queen 1 row movement
Occupied by Queen 1 and 2's movement	Occupied by Queen 1 and 2's movement	Queen 2 diagonal movement
Occupied by Queen 1 and 2's movement	Queen 2 placement	Occupied by Queen 1 and 2's movement

Diagram 2.3
Possible movement depicted if the rightmost middle column space was occupied by queen 2.

Queen 1 placement	Occupied by Queen 1 and 2's movement	Occupied by Queen 1 and 2's movement
Occupied by Queen 1 and 2's movement	Occupied by Queen 1 and 2's movement	Queen 2 placement
Queen 1 column movement	Queen 2 diagonal movement	Occupied by Queen 1 and 2's movement

Discuss the following:
a. Discuss how recursion is used in this project.

Recursion is used within this project to check and place a queen onto a table. We simulate each row in an n by n matrix, and check to see if it's possible for a queen to be placed there. Only if it has this possibility, does the function increment by one and call upon itself to find the location, Once the location is found, the method moves this value up the previous calls and moves onto the next queen.

b. Discuss why the expand algorithm is more efficient than the checknode algorithm.

The expand algorithm is more efficient than the checknode algorithm because of how expand() handles its recursive calls. Within the expand algorithm, we begin by assigning each i + 1 value to be a specified value from 0 to n-1. Afterwards, we check this value, and only if there is a possible solution does it make the recursive call to playQueens(). However, within the check node algorithm, we first check to see if there is a solution at i. If there could be a solution, and it is not the solution we need, then the system makes a recursive call for every single value from 0 to n-1 to see if it is the given solution to the case. In comparison, expand() only makes the recursive call when the solution has yet to be found, however checknode() makes the recursive call for every individual value.