A magic square is a $3 \times 3$ matrix of $\overset{\text{distinct}}{\vee}$ positive integers so that the sum of each diagonal, row, and column is equal, e.g.

| 10 | 3 | 8 | →21 |
|----|---|---|-----|
| 5  | 7 | 9 | →21 |
| 6  | 11| 4 | →21 |

↙ ↓ ↓ ↓ ↘
21   21   21   21    21

| 2 | 7 | 6 | →15 |
|---|---|---|-----|
| 9 | 5 | 1 | →15 |
| 4 | 3 | 8 | →15 |

↙ ↓ ↓ ↓ ↘
15   15   15   15    15

(the numbers need not be consecutive)

Suppose we wanted to create a state machine to find a magic square whose rows, columns, and diagonals all sum to integer $k$ (e.g. $k=21$ for the above left square, while $k=15$ for the above right).

Create a state machine $M = (Q, \Sigma, \Delta, q_0, F)$ with branching factor $b = 9$, maximum depth $m = 3k$, and solution depth $d = 3k$.

Which strategy should you use with this machine: BFS or DFS? Why?