

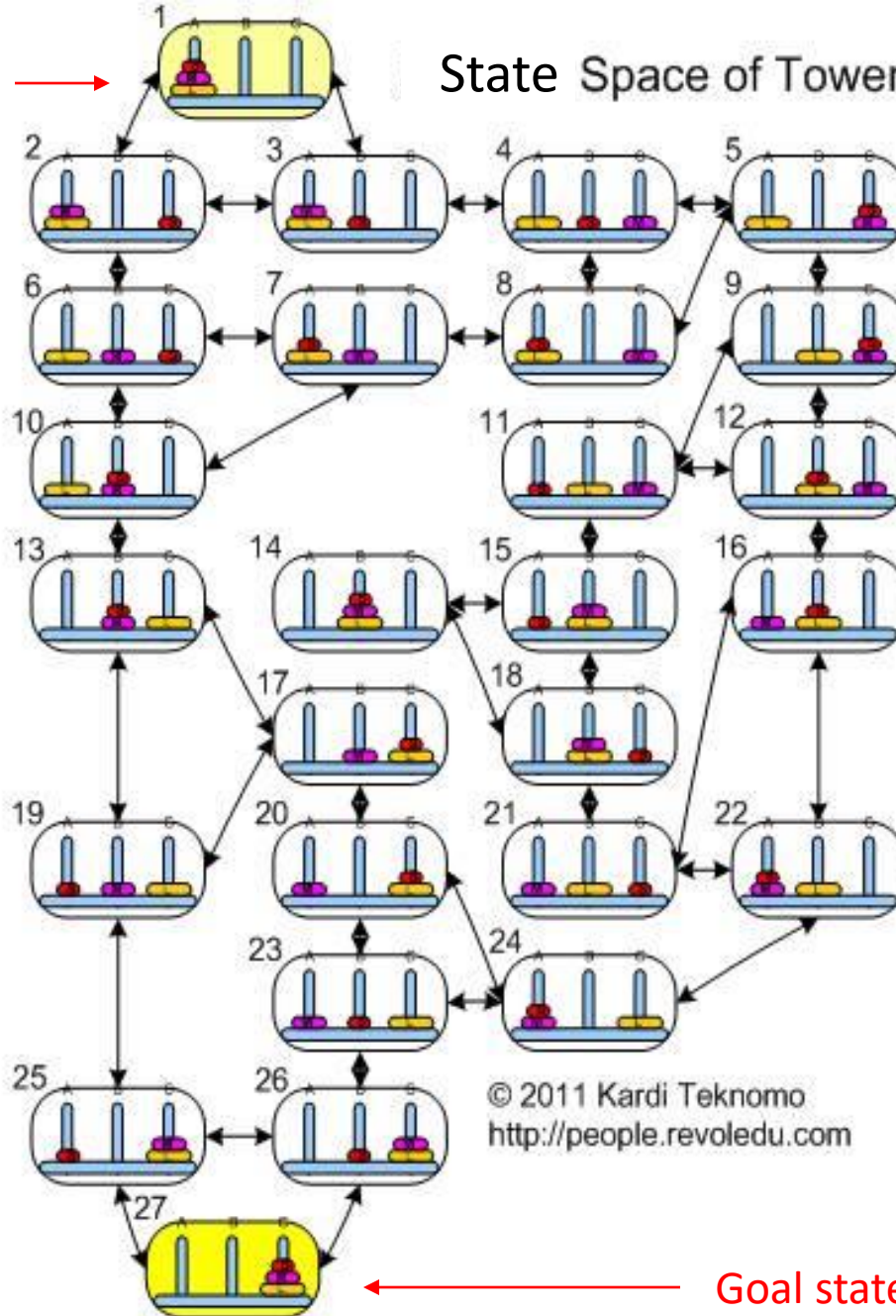
# State Space & Breadth-First Search

CSX3009 Algorithm Design

Initial state



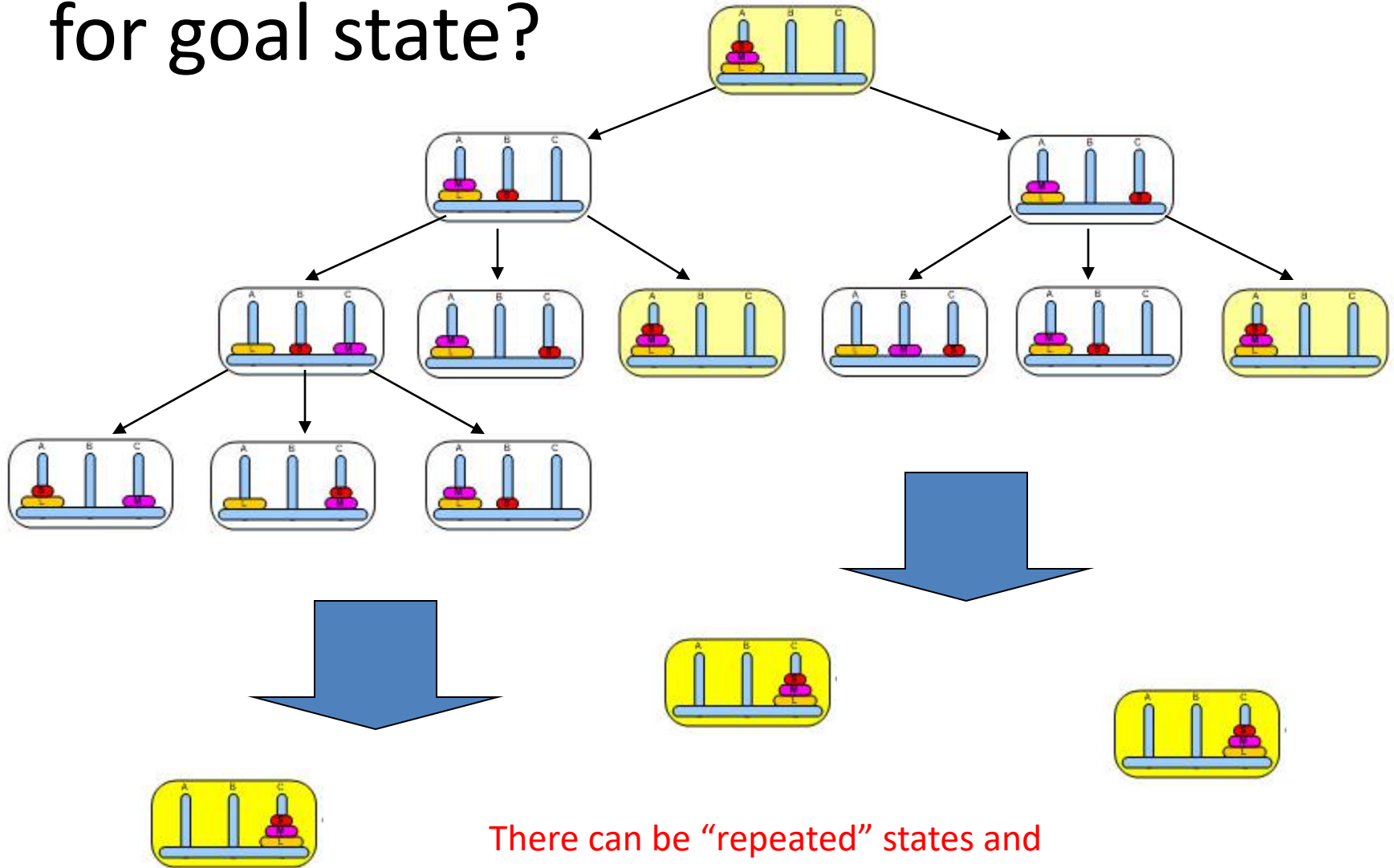
## State Space of Tower of Hanoi (3 disks)



Goal state

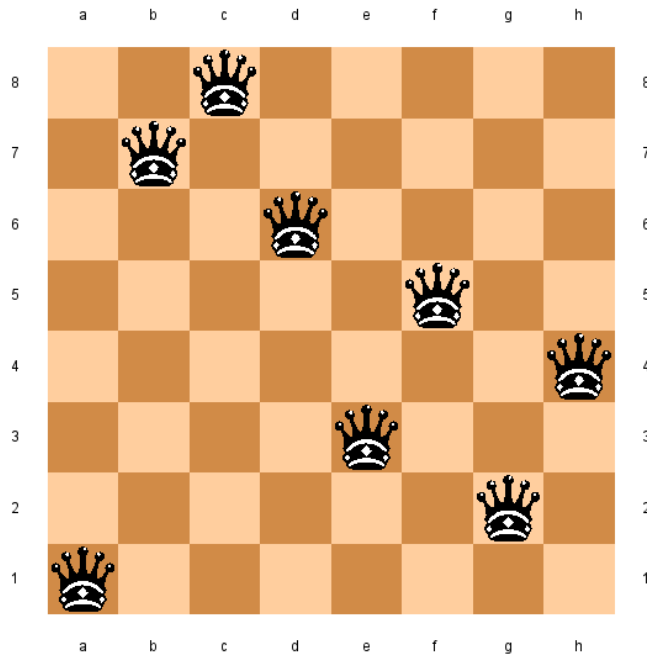
If you've never known about Tower of Hanoi problem, learn about it briefly [here](#).

# How to search for goal state?



# Modeling a problem

- N queens



## Complete-State Formation:

- **States:** any arrangement of 8 queens on the board
- **Actions:** move a queen to any square
- **Goal test:** 8 queens on the board, none attacked

Each queen can take any of the 64 positions  
→  $64^8$  states with 8 queens ( $2.81 \times 10^{14}$ )

# Incremental Formulation I

- **States:** Any arrangement of 0 to 8 queens on the board
- **Initial state:** No queens
- **Actions:** Add queen in empty square

First queen: 64 choices of position

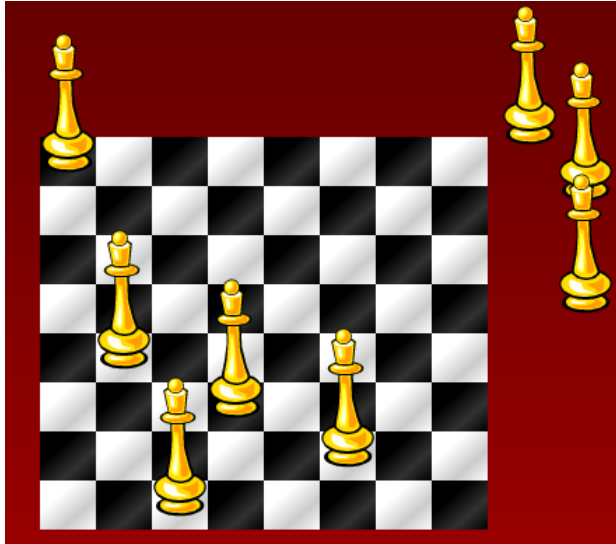
Second queen: 63 choices of position

...

Total number of possible states :  $64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57$

**$1.785 \times 10^{14}$  states**

# Incremental Formulation II



**States:** Any arrangement of 0 to 8 queens on the board

**Initial state:** No queens

**Actions:** Add queen in *leftmost empty column* such that is not attacking other queens

**First queen :** 8 choices of rows

**Second queen :** 5 or 6 choices of rows  
(must not be the same row as the first,  
must not be in diagonal with the first)

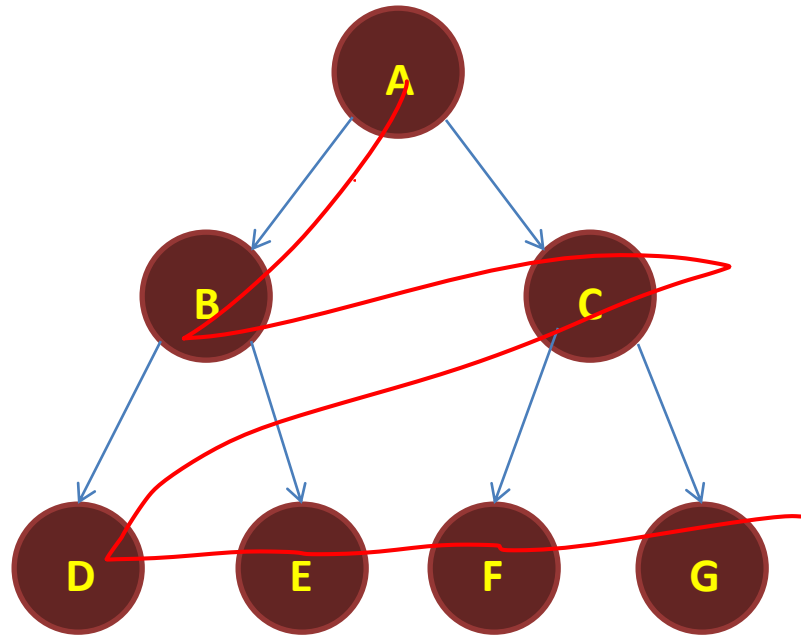
...

A lot fewer choices for the rest of the queens

**2057 states**

# Breadth First Search

Analogous to the diagram in page 3  
(searching for goal state of Tower of Hanoi)



**Click through animation to see  
sequence of states being searched**

# Algorithm structure of BFS

`s = initial_state`

**while** not Goal(s)

**for each** successor\_state x of s

        enqueue(x)

    s = dequeue()



# 4-Queens

