

# Game of 2048

J.-S. Roger Jang (張智星)

[jang@mirlab.org](mailto:jang@mirlab.org)

<http://mirlab.org/jang>

MIR Lab, CSIE Dept.

National Taiwan University

# Introduction to 2048

---

## ○ About 2048

- Try out: <http://2048game.com>
- Developed by 19-year-old Italian web developer Gabriele Cirulli
- Release date: March 9, 2014
- Goal: for an element to reach 2048
- Reference on Wiki: [English](#), [Chinese](#)

## ○ Your mission

- Write a program to find an input sequence to reach the goal based on the concept of stacks for DFS (depth-first search).

# Properties of 2048

---

- About solution
  - May not be solvable
  - May have multiple solutions
- The behavior of the next map is implementation dependent
  - How to combine same-value elements
  - Probability of generating 2 and 4
  - TA will give you a function to generate the next state

# Solution to 2048

- Exhaustive search
  - $4^m \rightarrow$  Impossible for a large  $m$ !
- Heuristic search
  - Conservative: Maximize 2's coverage
  - Aggressive: Maximize
    - Number of merges
    - Sum of merges
  - Many more strategies on the web
- Search based on maze traversal
  - DFS (depth-first search)  $\rightarrow$  Preferred
  - BFS (breadth-first search)  $\rightarrow$  Need much more memory

Quiz!

# Heuristic Search: Examples

---

- Examples

## Pseudo code for DFS Search

Push the initial map to stack

If size of stack  $> 0$  {

Pop stack to have a map A

If A fulfils the requirement, **return A**

Create p ( $p \leq 4$ ) viable maps after each action

Push these p maps to stack

}

P could be zero.

**No solution exists**

## More about 2048 HW

---

- TA will provide nextMap() for you to use
  - map2=nextMap(map, action)
    - map: current map
    - action: 0: east, 1: south, 2: west, 3: north
    - map2: next maps
  - Details to be provided later
- To speed up, we may
  - Start from a half-filled map
  - Allow multiple merges in nextMap()
- Either a stack or recursive backtracking will do the job
- Limits of time and memory to be determined later