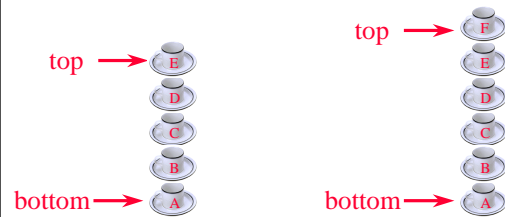


Stacks



- Linear list.
- One end is called **top**.
- Other end is called **bottom**.
- Additions to and removals from the **top** end only.

Stack Of Cups



- Add a cup to the stack.
- Remove a cup from new stack.
- A stack is a LIFO list.

The Interface Stack

```
public interface Stack
{
    public boolean empty();
    public Object peek();
    public void push(Object theObject);
    public Object pop();
}
```

Parentheses Matching

- $((a+b)*c+d-e)/(f+g)-(h+j)*(k-l)/(m-n)$
 - Output pairs (u,v) such that the left parenthesis at position **u** is matched with the right parenthesis at **v**.
 - (2,6) (1,13) (15,19) (21,25) (27,31) (0,32) (34,38)
- $(a+b)*((c+d)$
 - (0,4)
 - right parenthesis at 5 has no matching left parenthesis
 - (8,12)
 - left parenthesis at 7 has no matching right parenthesis

Parentheses Matching

- scan expression from left to right
- when a left parenthesis is encountered, add its position to the stack
- when a right parenthesis is encountered, remove matching position from stack

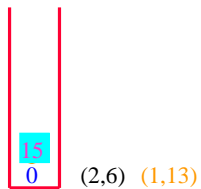
Example

- (((a+b)*c+d-e)/(f+g)-(h+j)*(k-l))/(m-n)



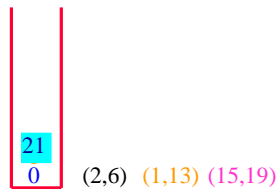
Example

- (((a+b)*c+d-e)/(f+g)-(h+j)*(k-l))/(m-n)



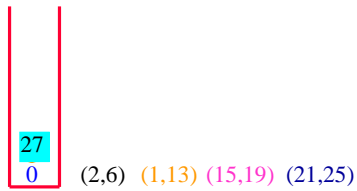
Example

- (((a+b)*c+d-e)/(f+g)-(h+j)*(k-l))/(m-n)



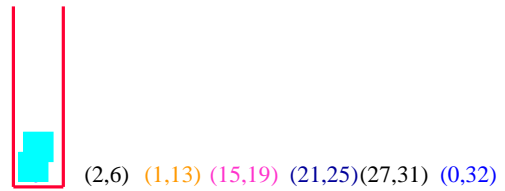
Example

- $((a+b)*c+d-e)/(f+g)-(h+j)*(k-l)/(m-n)$



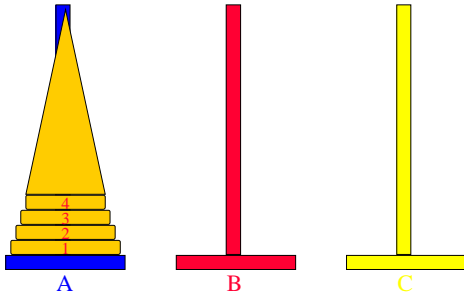
Example

- $((a+b)*c+d-e)/(f+g)-(h+j)*(k-l)/(m-n)$



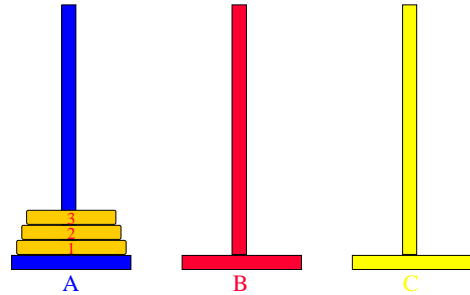
- and so on

Towers Of Hanoi/Brahma

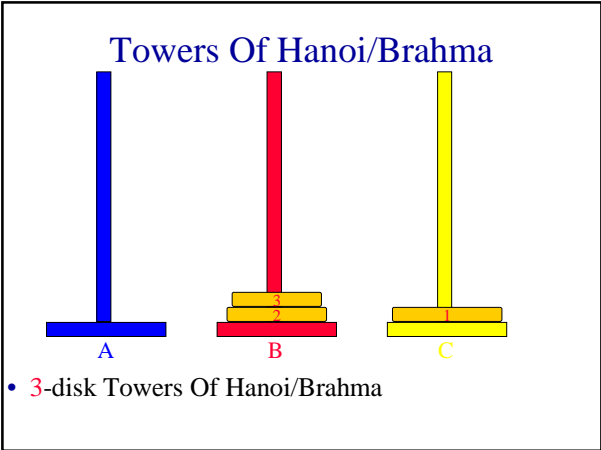
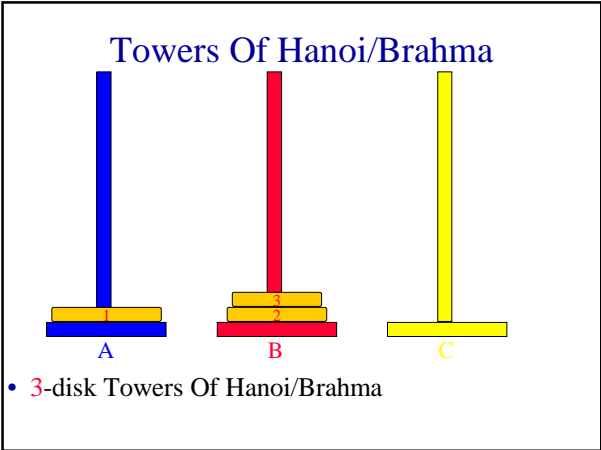
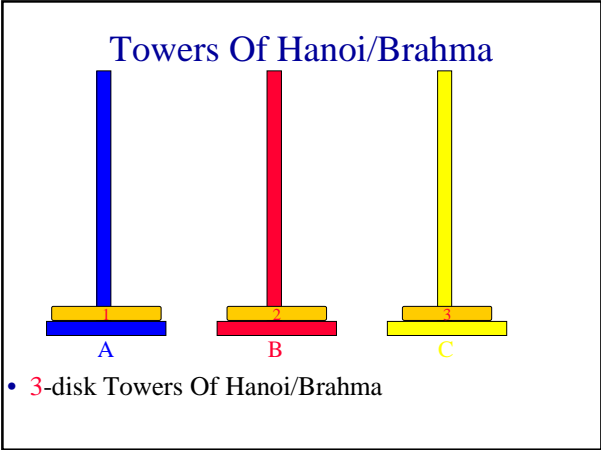
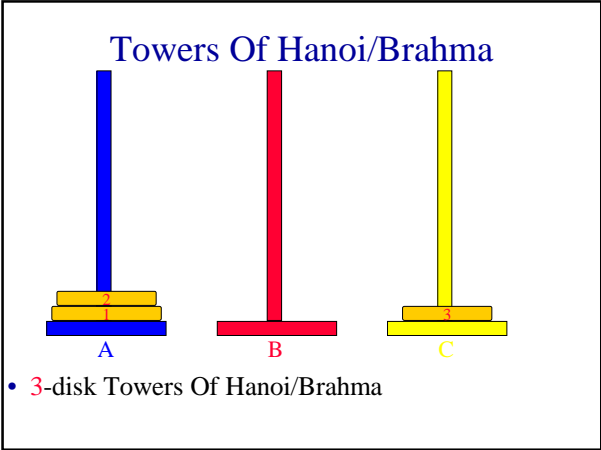


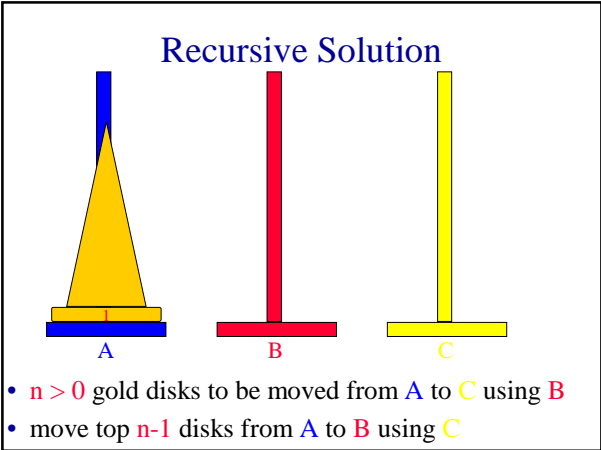
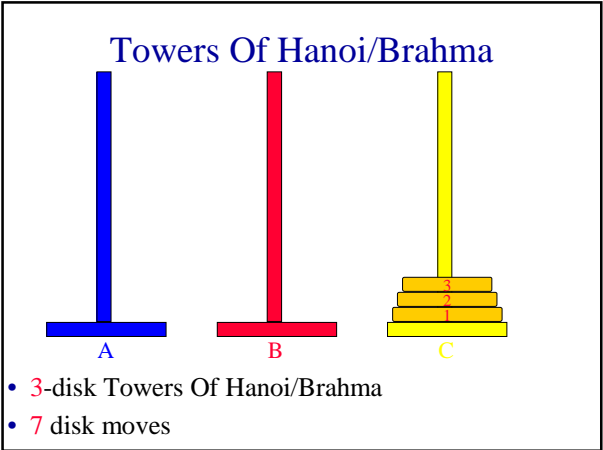
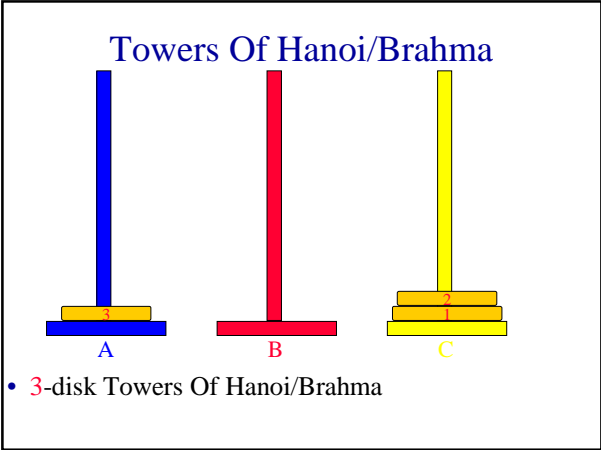
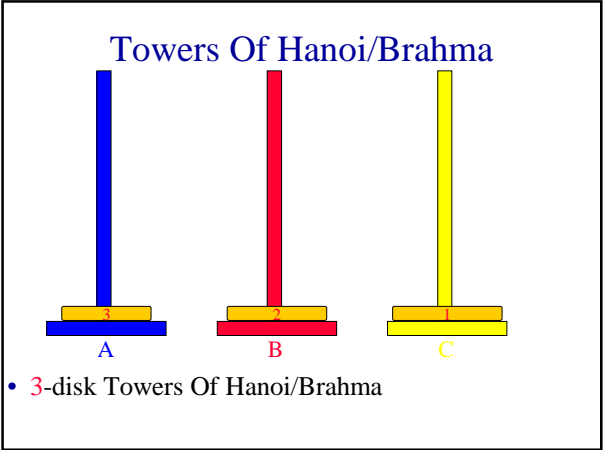
- 64 gold disks to be moved from tower A to tower C
- each tower operates as a stack
- cannot place big disk on top of a smaller one

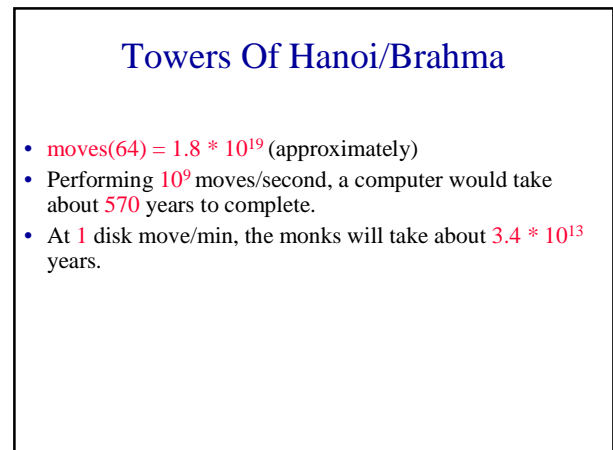
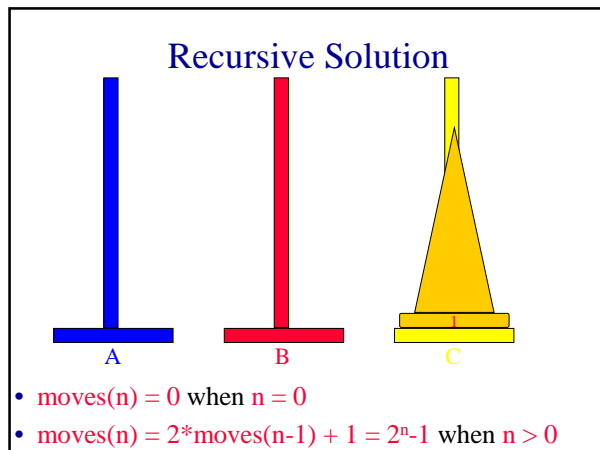
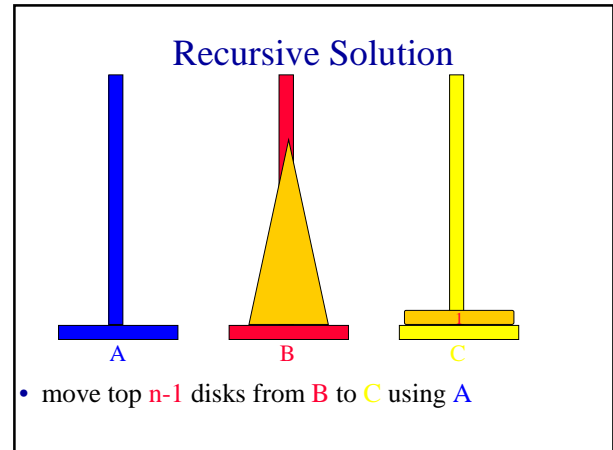
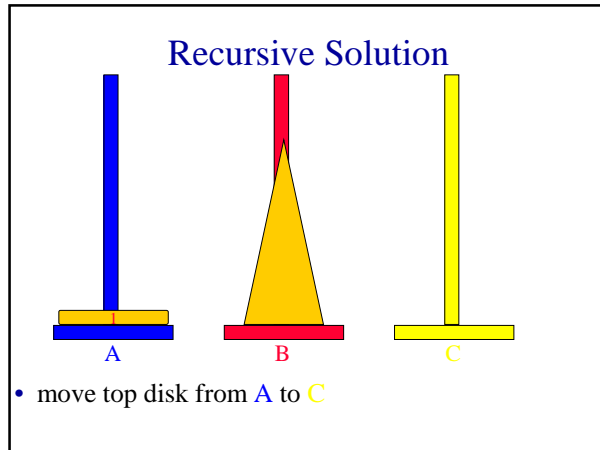
Towers Of Hanoi/Brahma



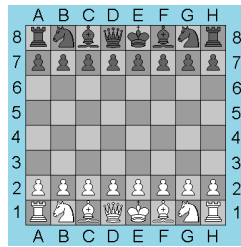
- 3-disk Towers Of Hanoi/Brahma





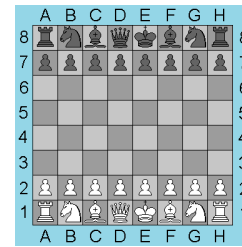


Chess Story



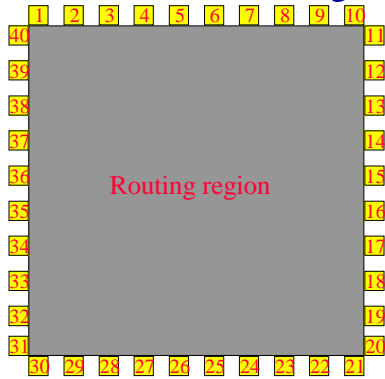
- 1 grain of rice on the first square, 2 for next, 4 for next, 8 for next, and so on.
- Surface area needed exceeds surface area of earth.

Chess Story

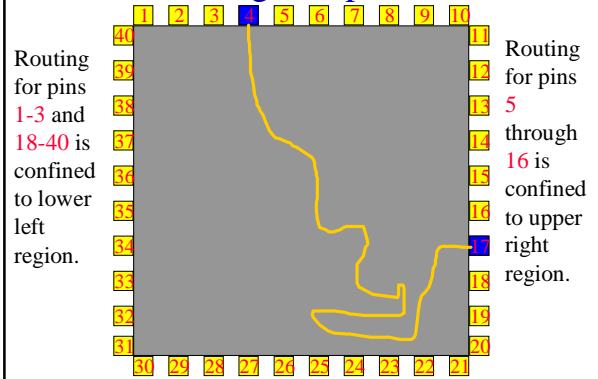


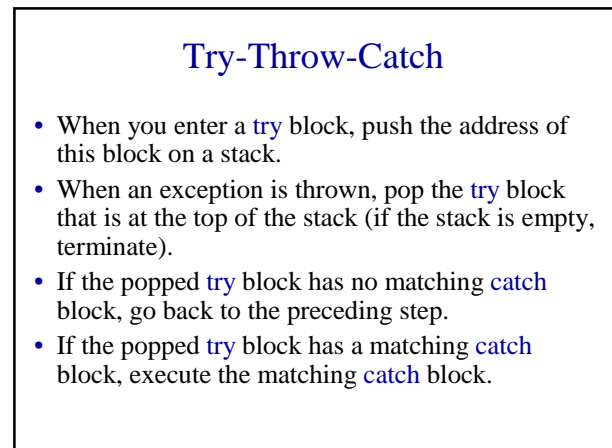
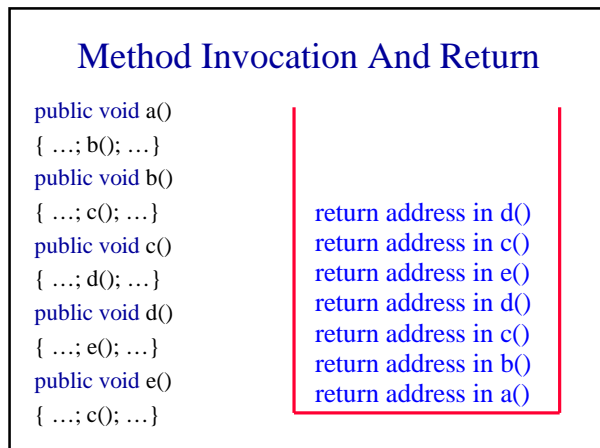
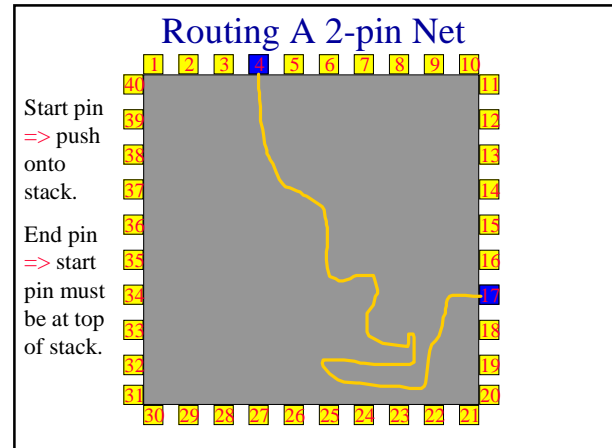
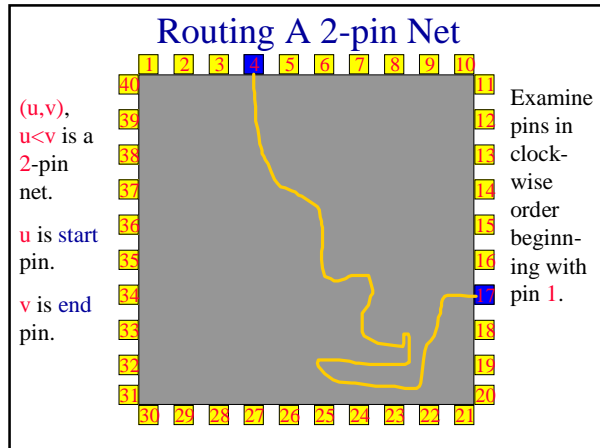
- 1 penny for the first square, 2 for next, 4 for next, 8 for next, and so on.
- $\$3.6 * 10^{17}$ (federal budget $\sim \$2 * 10^{12}$).

Switch Box Routing

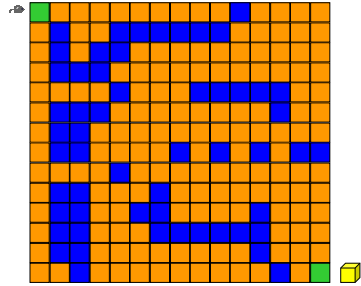


Routing A 2-pin Net

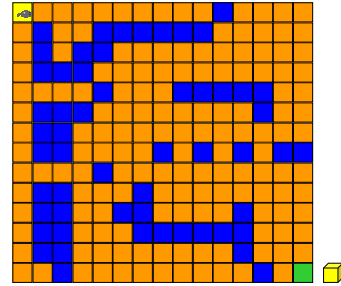




Rat In A Maze

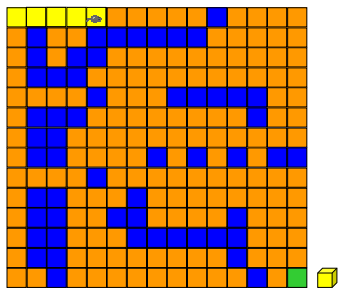


Rat In A Maze



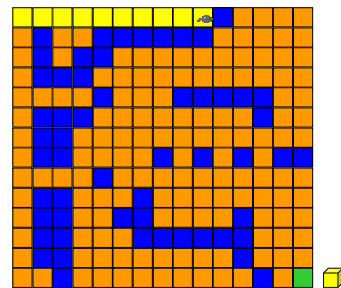
- Move order is: **right**, **down**, **left**, **up**
- Block positions to avoid revisit.

Rat In A Maze



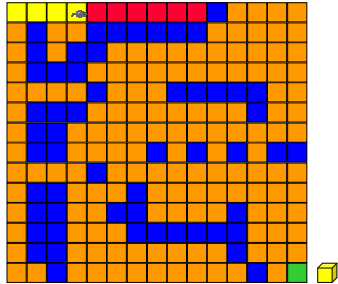
- Move order is: **right**, **down**, **left**, **up**
- Block positions to avoid revisit.

Rat In A Maze



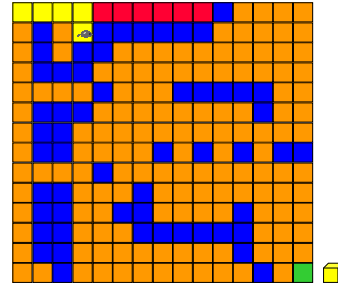
- Move backward until we reach a square from which a forward move is possible.

Rat In A Maze



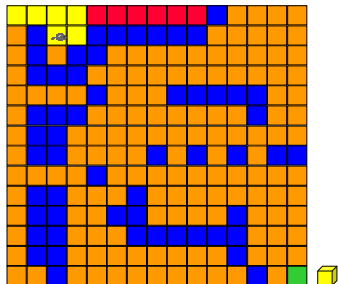
- Move down.

Rat In A Maze



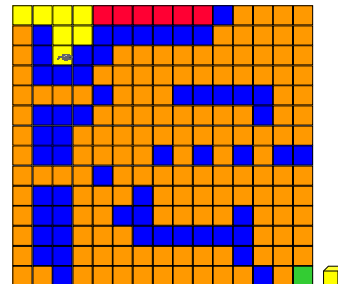
- Move left.

Rat In A Maze



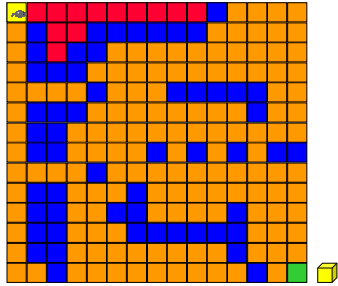
- Move down.

Rat In A Maze



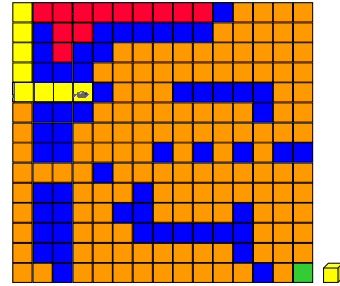
- Move backward until we reach a square from which a forward move is possible.

Rat In A Maze



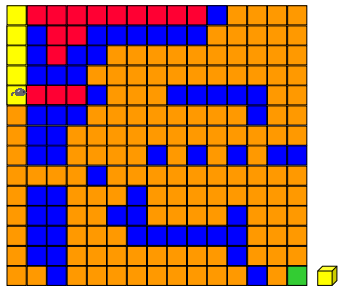
- Move backward until we reach a square from which a forward move is possible.
- Move downward.

Rat In A Maze



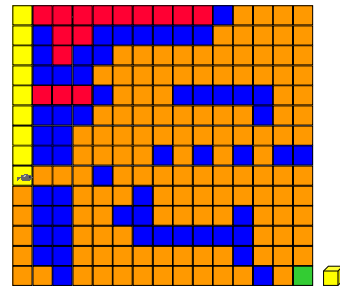
- Move right.
- Backtrack.

Rat In A Maze



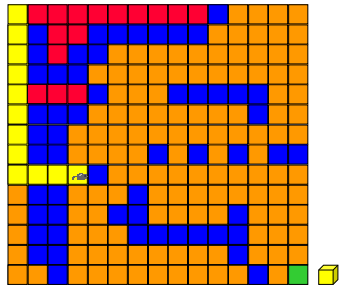
- Move downward.

Rat In A Maze



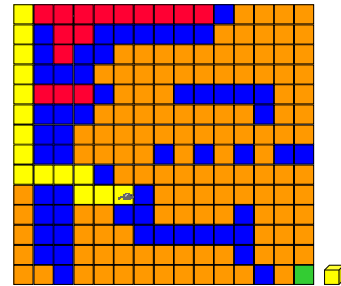
- Move right.

Rat In A Maze



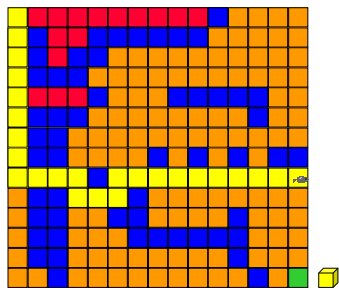
- Move one down and then right.

Rat In A Maze



- Move one up and then right.

Rat In A Maze



- Move down to exit and eat cheese.
- Path from maze entry to current position operates as a stack.