Picking up from Last Time
**Goal:** make a TM that checks if there are the same number of 0s and 1s in a string.

**Step 1:** sort the digits of the string

**Step 2:** use the TM we already made that can check if a string is in the form $0^n1^n$
Based on what we want this TM to do, what should this transition say?

A. \(0 \rightarrow 0, \text{R}\)
B. \(0 \rightarrow 1, \text{R}\)
C. \(0 \rightarrow 0, \text{L}\)
D. \(0 \rightarrow 1, \text{L}\)
E. None of these, or two or more of these.

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, D, or E.
0 \rightarrow 0, R

0^* \rightarrow 1 \rightarrow 1, R

0^*1^* \rightarrow 1 \rightarrow 1, R

0 \rightarrow ?, ?

... 0 0 1 1 0 0 ...
0 \rightarrow 0, R

1 \rightarrow 1, R

0 \rightarrow 1, L

0 \rightarrow 1, R

... 0 0 1 1 0 0 ...

0 → 0, R

0* → 0, R

1 → 1, R

0*1* → 1, R

1 → 1, R

0 → 1, L

1 → 0, L

Fix 01

0 0 1 0 1 0 1 0
\[
\begin{align*}
0 & \rightarrow 0, R \\
0^* & \rightarrow 1, R \\
0^*1^* & \rightarrow 1, R \\
0^*1^* & \rightarrow 0, L \\
0 & \rightarrow 1, L \\
1 & \rightarrow 1, L \\
1 & \rightarrow 0, L \\
\text{Go Home} & \rightarrow 0, L \\
\text{Fix 01} & \rightarrow 0, L \\
\end{align*}
\]
\begin{itemize}
  \item \textbf{start} \quad 0 \rightarrow 0, R
  \item 1 \rightarrow 1, R
  \item \textbf{0*} \quad 0 \rightarrow 1, L
  \item 1 \rightarrow 1, L
  \item \textbf{0*1*} \quad 0 \rightarrow 1, L
  \item \textbf{Fix 01} \quad 1 \rightarrow 0, L
  \item \textbf{Go Home} \quad 1 \rightarrow 1, L
\end{itemize}

\begin{array}{cccccccc}
... & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0...
\end{array}
$0 \rightarrow 0, R$

Start

$0^* 

1 \rightarrow 1, R$

$0^*1^* 

1 \rightarrow 1, R$

$0 \rightarrow 1, L$

$1 \rightarrow 0, L$

Fix

01

Go

Home

$0 \rightarrow 0, L$

$1 \rightarrow 1, L$

... 0 0 1 0 1 0 1 0 ...

...
\begin{align*}
0 \rightarrow 0, R & \quad 0^* \\
1 \rightarrow 1, R & \quad 0^*1^* \\
\square \rightarrow \square, R & \quad \text{Fix} 01 \\
0 \rightarrow 0, L & \quad \text{Go Home} \\
1 \rightarrow 1, L & \quad 1 \rightarrow 0, L \\
0 \rightarrow 1, L & \quad 1 \rightarrow 1, R \\
\end{align*}
0 → 0, R
1 → 1, R

0* → 0*, R

0*1* → 0*1*, R
1 → 1, R

□ → □, R

0 → 0, L
1 → 1, L
Go Home

1 → 0, L

0 → 1, L
Fix 01

Go Home
0 → 0, R

□ → □, R

0 → 0, L
1 → 1, L

0* → 0*
1 → 1, R

0*1* → 0*1*
1 → 1, R

0 → 1, L

Go Home

1 → 0, L

Fix 01

... 0 0 0 0 1 1 1 0 ...
0 → 0, R
0* → 0*, R
0* → 0, L
1 → 0, L

1 → 1, R
0*1* → 0*1*, R
1 → 1, R

□ → □, R

0 → 1, L

Go Home

Fix 01

00001110
\[ \begin{align*}
0 & \rightarrow 0, L \\
1 & \rightarrow 1, L \\
0 & \rightarrow 0, R \\
1 & \rightarrow 1, R \\
\end{align*} \]
\[ 0 \rightarrow 0, R \]
\[ 1 \rightarrow 1, R \]
\[ \square \rightarrow \square, R \]
\[ 0 \rightarrow 0, L \]
\[ 1 \rightarrow 1, L \]
\[ 1 \rightarrow 0, L \]
\[ 0 \rightarrow 1, L \]

- **0**
- **01**
- **Fix 01**

- **Start**
\(0 \rightarrow 0, R\)
\(0*\)
\(1 \rightarrow 1, R\)
\(0*1*\)
\(1 \rightarrow 1, R\)
\(\square \rightarrow \square, R\)
\(0 \rightarrow 0, L\)
\(0 \rightarrow 0, L\)
\(1 \rightarrow 1, L\)
\(1 \rightarrow 0, L\)
\(\text{Go Home}\)
\(\text{Fix 01}\)

\(\ldots\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ \ldots\)
\[
\begin{align*}
0 &\rightarrow 0, R \\
0* &\rightarrow \square, R \\
\square &\rightarrow \square, R \\
0 &\rightarrow 0, L \\
1 &\rightarrow 1, L \\
0*1* &\rightarrow 1 \rightarrow 1, R \\
1 &\rightarrow 1, R \\
0 &\rightarrow 1, L \\
1 &\rightarrow 0, L
\end{align*}
\]
\[ (0 \rightarrow 0, R) \quad (1 \rightarrow 1, R) \quad (0 \rightarrow 1, L) \quad (1 \rightarrow 0, L) \]

- Start state: \( 0^* \)
- Fix 01 state: \( 0^*1^* \)
- Go Home state: \( \square \)

Input sequence: \( \ldots 000010101 \ldots \)
\[ s \rightarrow t \rightarrow r \]

\[ 0 \rightarrow 0, R \]
\[ 1 \rightarrow 1, R \]
\[ 0 \rightarrow 1, L \]
\[ 1 \rightarrow 0, L \]

\[ \square \rightarrow \square, R \]
\[ 0 \rightarrow 1, L \]

\[ \text{Go Home} \]

\[ \text{Fix 01} \]
0 → 0, R

0* → 0, R

1 → 1, R

0 → 1, R

0*1* → 1 → 1, R

1 → 0, L

Go Home → Go Home

0 → 0, L

1 → 1, L

□ → □, R

0 → 1, L

Fix 01 → Fix 01

1 → 0, L
... 0 0 0 0 1 0 1 1 ...
\[ 0 \rightarrow 0, R \]
\[ 1 \rightarrow 1, R \]
\[ \square \rightarrow \square, R \]
\[ 0 \rightarrow 1, L \]
\[ 1 \rightarrow 0, L \]

Diagram:

- Start state: \( 0^* \)
- Transitions:
  - \( 0 \rightarrow 0, R \)
  - \( 1 \rightarrow 1, R \)
  - \( \square \rightarrow \square, R \)
  - \( 1 \rightarrow 0, L \)
  - \( 0 \rightarrow 1, L \)

States:
- \( 0^* \)
- Go Home
- Fix 01
\[ 0 \rightarrow 0, \text{R} \]
\[ 0 \rightarrow 0, \text{L} \]
\[ 1 \rightarrow 1, \text{R} \]
\[ 1 \rightarrow 0, \text{L} \]

- Start

- Go Home:
  - 0 → 0, R
  - 1 → 1, L

- Fix 01:
  - 0 → 1, L

- Transition:
  - □ → □, R
  - 1 → 1, R

Input:
- ... 0 0 0 0 1 0 1 ...
0 → 0, R
0* → 0*, R
1 → 1, R
0*1* → 0*, R
1 → 1, R
□ → □, R

0 → 0, L
1 → 1, L
Go Home 

1 → 0, L
0 → 1, L
Fix 01 

... 0 1 0 1 ...
\[ \begin{array}{cccccc}
0 & \rightarrow & 0, & R \\
\square & \rightarrow & \square, & R \\
0 & \rightarrow & 0, & L \\
1 & \rightarrow & 1, & L \\
\hline
0* & \rightarrow & 1, & R \\
0*1* & \rightarrow & 1, & R \\
\hline
Go \text{Home} & \rightarrow & 1, & L \\
Fix \text{01} & \rightarrow & 0, & R \\
\end{array} \]

\[ \cdots \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad \cdots \]
\[0 \rightarrow 0, R\]
\[0 \rightarrow 0, L\]
\[1 \rightarrow 1, R\]
\[1 \rightarrow 0, L\]

Start state:

- \[0^*\] with transitions:
  - \[0 \rightarrow 0, R\]
  - \[1 \rightarrow 1, R\]

- \[\square \rightarrow \square, R\]

- \[\rightarrow \square, R\]

Halting states:

- \[01\] with transitions:
  - \[0 \rightarrow 1, L\]
  - \[1 \rightarrow 1, L\]
0 → 0, R
1 → 1, R

0* → 0*
1 → 1, R

□ → □, R

0 → 1, L

0 → 0, L
1 → 1, L

Go Home → 0*
1 → 0, L

Fix 01 → 0*
1 → 1, R

... 0 0 0 0 0 1 1 ...
Our ultimate goal here was to sort everything so we could hand it off to the machine to check for $0^n1^n$. Let's rewind the tape head back to the start.
To Start

0 → 0, R

0 → 0, L
1 → 1, L

0*1*

1 → 1, R

0*1*

1 → 1, R

Go Home

0 → 0, R

1 → 1, R

0 → 1, L
1 → 0, L

Fix 01

0 → 1, L

To Start

... 0 0 0 0 0 1 1 ...
To Start

0* → 0, R

0 → 0, R

1 → 0, L

0 → 0, L

1 → 1, L

Go

Home

... 0 0 0 0 0 1 1 ...
\[
\begin{array}{ll}
0 \rightarrow 0, L & 1 \rightarrow 1, L \\
1 \rightarrow 0, L & 0 \rightarrow 1, L \\
1 \rightarrow 1, L & 0 \rightarrow 1, L \\
0 \rightarrow 0, R & 1 \rightarrow 1, R \\
0 \rightarrow 0, R & 1 \rightarrow 0, L \\
1 \rightarrow 1, R & 0 \rightarrow 1, L \\
\end{array}
\]
0 \rightarrow 0, \text{L} \\
1 \rightarrow 1, \text{L}

To Start

\[ \begin{array}{c}
0 \rightarrow 0, \text{L} \\
1 \rightarrow 1, \text{L}
\end{array} \]

\[ \begin{array}{c}
\square \rightarrow \square, \text{R} \\
\square \rightarrow \square, \text{L}
\end{array} \]

\text{Start } 0^n1^n

0 \rightarrow 0, \text{R} \\
1 \rightarrow 1, \text{R}

0^* \\
1 \rightarrow 1, \text{R}

\[ \begin{array}{c}
0 \rightarrow 0, \text{R} \\
\square \rightarrow \square, \text{R}
\end{array} \]

\[ \begin{array}{c}
0 \rightarrow 1, \text{L} \\
1 \rightarrow 1, \text{L}
\end{array} \]

\text{Go Home}

\[ \begin{array}{c}
0 \rightarrow 0, \text{L} \\
1 \rightarrow 1, \text{L}
\end{array} \]

\[ \begin{array}{c}
1 \rightarrow 0, \text{L} \\
0 \rightarrow 1, \text{L}
\end{array} \]

\text{Fix 01}
This is just a placeholder. Imagine snapping in the entire TM for $0^n1^n$ into this diagram, putting the start state in the dashed area.
TM Subroutines

- A **TM subroutine** is a Turing machine that, instead of accepting or rejecting an input, does some sort of processing job.
- TM subroutines let us compose larger TMs out of smaller TMs, just as you'd write a larger program using lots of smaller helper functions.
- Here, we saw a TM subroutine that sorts a sequence of 0s and 1s into ascending order.
TM Subroutines

• Typically, when a subroutine is done running, you have it enter a state marked “done” with a dashed line around it.

• When we're composing multiple subroutines together – which we'll do in a bit – the idea is that we'll snap in some real state for the “done” state.
TM Arithmetic

• Let's design a TM that, given a tape that looks like this:

  ... 1 3 7 4 2 ...

  ... 1 7 9 0 0 ...

ends up having the tape look like this:

• In other words, we want to build a TM that can add two numbers.
TM Arithmetic

• There are many ways we could in principle design this TM.

• We're going to take the following approach:
  - First, we'll build a TM that increments a number.
  - Next, we'll build a TM that decrements a number.
  - Then, we'll combine them together, repeatedly decrementing the second number and adding one to the first number.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  - the tape head points at the start of a number,
  - there is at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there are at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Let's begin by building a TM that increments a number.

We'll assume that

- the tape head points at the start of a number,
- there is at least two blanks to the left of the number, and
- that there's at least one blank at the start of the number.

The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.
• We'll assume that
  – the tape head points at the start of a number,
  – there is at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.
• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

- Let's begin by building a TM that increments a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there are at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.
- The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there are at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  - the tape head points at the start of a number,
  - there is at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there is at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

- Let's begin by building a TM that increments a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there are at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.
- The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  - the tape head points at the start of a number,
  - there is at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  − the tape head points at the start of a number,
  − there is at least two blanks to the left of the number, and
  − that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there is at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  - the tape head points at the start of a number,
  - there is at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there is at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.
• We'll assume that
  - the tape head points at the start of a number,
  - there is are at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.
• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

• Let's begin by building a TM that increments a number.

• We'll assume that
  – the tape head points at the start of a number,
  – there is are at least two blanks to the left of the number, and
  – that there's at least one blank at the start of the number.

• The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

- Let's begin by building a TM that increments a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there is at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.
- The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

- Let's begin by building a TM that increments a number.
- We'll assume that
  - the tape head points at the start of a number,
  - there are at least two blanks to the left of the number, and
  - that there's at least one blank at the start of the number.
- The tape head will end at the start of the number after incrementing it.
Incrementing Numbers

go to the end of the number;

while (the current digit is 9) {
    set the current digit to 0;
    back up one digit;
}

increment the current digit;

go to the start of the number;
... 1 2 9 9 ...
start

0 → 0, R
1 → 1, R
...
9 → 9, R

To

End

... 1 2 9 9 ...

...
0 → 0, R
1 → 1, R
... 
9 → 9, R
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

... 1 2 9 9 ...

...
0 → 0, R
1 → 1, R
...
9 → 9, R
To End

0 → 0, R
1 → 1, R
...
9 → 9, R
0 → 0, R
1 → 1, R
...
9 → 9, R

start

To End

□ → □, L

... 1 2 9 9 ...

...
Based on we want this TM to do, what should this transition say?

A. 0 → 9, R
B. 0 → 9, L
C. 9 → 0, R
D. 9 → 0, L
E. None of these, or two or more of these.

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, D, or E.
0 \rightarrow 0, \text{R}
1 \rightarrow 1, \text{R}
\ldots
9 \rightarrow 9, \text{R}

\text{start}

\text{To End} \quad \square \rightarrow \square, \text{L}

\text{Wrap Nines}

? \rightarrow ?, ?

\ldots 1 2 9 9 \ldots
To End

0 → 0, R
1 → 1, R
... 
9 → 9, R

□ → □, L

Wrap Nines

9 → 0, L

... 1 2 9 9 ...
0 → 0, R
1 → 1, R
...
9 → 9, R

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

0 → 0, L
1 → 1, L
...
9 → 9, L

... 1 3 0 0 ...
start

0 → 0, R
1 → 1, R
...
9 → 9, R

To End

□ → □, L

Wrap Nines

9 → 0, L
0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

done!

□ → □, R

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

... 1 3 0 0 ...

...
0 → 0, R
1 → 1, R
...
9 → 9, R

start

To End → □ → □, L

Wrap Nines → 9 → 0, L
0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

done!

□ → □, R

Back Home
0 → 0, L
1 → 1, L
...
9 → 9, L
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

9 → 0, L

Start

Back Home

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

done!

0 → 0, L
1 → 1, L
...
9 → 9, L

...
To End

Wrap Nines

done!

Start

0 → 0, R
1 → 1, R
...
9 → 9, R

□ → □, L

9 → 0, L
0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

□ → □, R

0 → 0, L
1 → 1, L
...
9 → 9, L

……
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

□ → □, L
9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

done!

□ → □, R

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

… 1 0 0 2 …
0 → 0, R
1 → 1, R
...
9 → 9, R

□ → □, L

9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

□ → □, R

0 → 0, L
1 → 1, L
...
9 → 9, L

...
0 → 0, R
1 → 1, R
...
9 → 9, R

□ → □, L
□ → □, R

To
End

Wrap
Nines

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

9 → 0, L

done!

Back
Home

0 → 0, L
1 → 1, L
...
9 → 9, L

... 1 0 0 2 ...

...
To End

\[ \begin{align*}
0 & \rightarrow 0, \text{ R} \\
1 & \rightarrow 1, \text{ R} \\
\ldots \\
9 & \rightarrow 9, \text{ R}
\end{align*} \]

Wrap Nines

\[ \begin{align*}
\square & \rightarrow \square, \text{ L} \\
9 & \rightarrow 0, \text{ L}
\end{align*} \]

Back Home

\[ \begin{align*}
0 & \rightarrow 1, \text{ L} \\
1 & \rightarrow 2, \text{ L} \\
2 & \rightarrow 3, \text{ L} \\
\ldots \\
8 & \rightarrow 9, \text{ L} \\
9 & \rightarrow 9, \text{ L}
\end{align*} \]

done!

\[ \begin{align*}
\square & \rightarrow \square, \text{ R}
\end{align*} \]
0 → 0, R
1 → 1, R
...
9 → 9, R

□ → □, L

9 → 0, L
0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

□ → □, R

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

... 1 0 0 3 ...

...
0 → 0, R
1 → 1, R
...
9 → 9, R

start

To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

0 → 0, R
1 → 1, R
...
9 → 9, L

... 1 0 0 3 ...
0 → 0, R
1 → 1, R
...
9 → 9, R

start

To End

□ → □, L

Wrap Nines

9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

done!

□ → □, R

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

1 0 0 3

...
Diagram showing a transition from 'start' to 'To End', 'Wrap Nines', 'Back Home', and 'done!' with specific states and transitions:

- Start: 0 → 0, R
- To End: 1 → 1, R
- Wrap Nines: 9 → 9, R
- Back Home: 0 → 1, L
- 1 → 2, L
- 2 → 3, L
- ... (repeating)
- ... (repeating)
- 9 → 9, L
- done!: □ → □, R

The input sequence shown is: ...
1 0 0 3 ...
...
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

□ → □, R

Back Home

□ → □, L
To End

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

9 → 0, L

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

□ → □, R

□ → □, L

... 9 9 9 ...

...
To End Wrap Nines

0 → 0, R
1 → 1, R
...
9 → 9, R

Wrap Nines

9 → 0, L
0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

Back Home

0 → 0, L
1 → 1, L
...
9 → 9, L

done!

start

... 9 0 0 ...

...
To End

Wrap Nines

Back Home

0 → 0, R
1 → 1, R
...
9 → 9, R

0 → 1, L
1 → 2, L
2 → 3, L
...
8 → 9, L

0 → 0, L
1 → 1, L
...
9 → 9, L

start

\[ \square \rightarrow \square, \ L \]

\[ \square \rightarrow \square, \ R \]

done!

... 0 0 0 ...

...
start

To End

Wrap Nines

done!

Back Home