Homework: make sure (1) you explain your solution (2) all handwriting is legible (3) your solution is concise.

Today: < >, deciding, halting, recognising.

Cheating with < >

What is an algorithm? Answer: DFA/NFA limit algorithms Turing Machines: good general algorithm

$L = \{0^n1^n\}$ can be recognised by a Turing Machine

What if a Turing Machine with countably infinite number of tapes

- type 1: If you promise me that your algorithm will only use finitely many tapes for "the algorithm" the rest for storage, then OK.
- type 2: The literal: $f : (Q \times \Gamma)^* \rightarrow (Q \times \Gamma \times \{L, R, S\})^*$ is too powerful.

§3.3 What is an algorithm?

- Graph --- is it connected?
- Graph --- is it 3-colourable?

So we need to describe each graph (inputs) as a string over some alphabet $\Sigma$. 
Want to describe Turing machines, list to sort, etc.

\[
\langle \text{Graph} \rangle = \langle \text{fixed description} \rangle
\]

\[
\text{Graph} : (V, E)
\]

\[
\begin{aligned}
&\quad 1 \rightarrow 2 \\
&\quad 1 \rightarrow 3 \\
&\quad 3 \rightarrow 1
\end{aligned}
\]

\[
\begin{aligned}
&\quad \#1 \rightarrow 2 \\
&\quad \#2 \rightarrow \#4 \\
&\quad R \#3 \\
&\quad \#2 \rightarrow \#4
\end{aligned}
\]

\[
\begin{aligned}
&\quad \#1 \rightarrow \#3
\end{aligned}
\]

\[
\begin{aligned}
\text{Assume vertices numbered }& \rightarrow h \\
\quad \Sigma = \{0, 1, 9, \# \}
\end{aligned}
\]

Describe edges as pairs

\[
\langle \text{Graph} \rangle \text{ the unique description}
\]

sometimes uniqueness is not an issue

sometimes it is...

Can we check, given \( \langle G \rangle \), if \( G \) is 3-colourable, with a

Turing machine?

\[
\text{Input }\begin{aligned}
&\quad 4 \#1 \#1 \#2 \#2 \#4 \#1 \#1 \#1 \#1 \#1
\end{aligned}
\]

High level description: Read \( h \) vertices. Have a list of

\[
\begin{aligned}
&\quad \text{colours } 1, 2, 3
\end{aligned}
\]

\[
\begin{aligned}
&\quad 4 \text{ vertices }\begin{aligned}
&\quad \text{cycle } 1 1 1 2 \\
&\quad \text{through } 1 1 2 3
\end{aligned}
\end{aligned}
\]

any value \( 3 3 3 3 \) \( \text{for each value of } 1, 2, 3 \) or each

vertex check if it is a

valid 3 colouring
Can you take \{JavaScript, Python, C\} and convert to a Turing machine algorithm?

Ans: Yes, yes with a Turing machine.

Given a Turing machine, and input, can you see what happens as you run the T.m. on the input?

\( \langle M, i \rangle \quad T.m. = M = (Q, \Sigma, \Gamma, q_0, \pi, \delta) \)

To input to the Turing machine: Some set

Assume: \( \Gamma = \{1, 2, 3, \ldots, \Sigma\} \)

"Standardized" \( \Sigma = \{1, \ldots, \Sigma\} \)

Turing machine inputs \( \Gamma = \{1, \ldots, \Sigma, \pi, \pi, \ldots, \} \)

\( \delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\} \)

\( \delta \text{ consists of } \delta(1,1), \delta(1,2), \ldots, \delta(1, \pi), \delta(2,1), \ldots \)

L, R

Let \( A = \{0, \ldots, \pi, \# \} \): write down each T.m.
Input $\Sigma = \{1, 2, 3\}$

```
1 \# 2 \# 1 \# 3 \# 2 \# 2 \# 3
```