L1. 2022-09-07

Practicalities.

email: marcvinyals@gmail.com.

Grading:

15% drills
25% tests
35% final (perhaps practice)
optional: project
bonus: bug reports.

Office Hours:

TBD. For now meet after class

* No class on thanksgiving week.
Will have extra time after some lectures.

Course Plan:

Why are some computational problems harder than others?

Examples of problems:

→ Given string $s \in \{0,1\}^*$, is number of 1s div. by 3?	easy
→ Given string, is it a palindrome? (i.e. ww^R)	
→ $s \in \{a,b,c\}^*$, is $\#a = \#b = \#c$?	
→ Given graph, is it connected?	
→ Given two graphs, are they isomorphic?	don't know
→ Given propositional formula, does it have a solution?	believed hard
→ Given graph, find largest clique.	
→ Given program, will it stop?	impossible

Often finding solution as hard as determining if exists.

e.g. SAT:

```
set first var. to true.
if ∃ solution: continue
o/w set to false
continue.
```

So we will work with decision problems:

Language $L \subseteq \Sigma^*$ known.

Task: given w, is $w \in L$?

What do we mean by computer?

Want universal model, indep. of technology.

Conj. (Church-Turing thesis) Turing Machines can do the same computations as any other process.

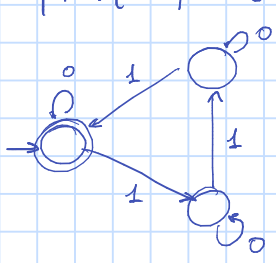
TMs are too complicated. As a warmup, we begin with finite automata.

Informally: state machine, only memory is current state.

Move between states with each character from input.

Represent with state diagram.

e.g. for $\{s : |i : s_i = 1| \equiv 0 \bmod 3\}$



Formally: tuple $(Q, \Sigma, \delta, q_0, F)$

Q: states (finite)
Σ: alphabet (finite)
δ: $Q \times \Sigma \to Q$ transition function
$q_0 \in Q$: initial state
$F \subseteq Q$: accepting states.