- $P$ vs. $NP$ (worth $\geq 10^6$ USD)

$P =$ polynomial time algorithms on a single/multi-tape Turing machine

$NP =$ non-deterministic

- $P$ contains all poly time algorithms in CPSC 320

- $NP$ includes 3SAT, SAT, 3COLOUR, PARTITION, ...
Breck: Monday 11/13 - 11/15

Monday Wednesday

HW 8 due Nov 16

In class HW 8 is up

Final exam! You can bring in 2 double-sided sheets 8.5"x11"

Some group HW 8 problems will be up today
Today

\[ P = \{ \text{Languages recognizable with polynomial time on a Turing machine} \} \]

Remark:

\[
\begin{align*}
P_{\text{T.M.}} &= P_{\text{Python}}
\end{align*}
\]

Warning: Our definitions will slightly vary from [Sip]
If $M$ is a Turing machine:

Initial Configuration (Step 1)

\[ a b b b a b a b a b a \]

2nd configuration (Step 2)

\[ a b b b a b a b a b a \]

\[ q \text{ reject} \]


Steps that $M$ takes on input $S$ is just the difference.

Step 1: initial config

Step 2

\[ \text{step 100} \]

(accept/reject)

\[ \# \text{steps} = 100 - 1 = 99 \]

\[ \text{time} = \# \text{steps} \]
Polynomial time:

If $f, g : \mathbb{N} \to \mathbb{R}$ and $g(n) > 0$ for $n$ sufficiently large, (i.e., $\exists n_0$ s.t. $g(n) > 0$ for all $n \geq n_0$), then we write

$$f(n) = \mathcal{O}(g(n))$$

= "big Oh" of $g(n)$

if there is a $C$ such that

$$|f(n)| \leq C g(n) \text{ for } n \text{ suff. large}$$
\[ f(n) \leq C \cdot g(n) \]

for all \( n \geq n_1 \)

If \( g(n) > 0 \) for \( n \) sufficiently large,

\[ \text{TIME}(g) \quad \text{or} \quad \text{TIME}(g(n)) \]
= \{ \text{languages, L, s.t. there is a Turing machine, M, that recognizes L that for each input of size n, M runs within time } O(g(n)) \}

(i.e. \exists C s.t. for all inputs of size n, M stops within time C g(n) for n sufficiently large)
\[ P = \text{PolyTime}_{\mathcal{T}_m} \]

\[ = \bigcup_{k=1, 2, \ldots} \text{TIME} \left( n^k \right) \]

Remark:

Could also say

\[ P = \left\{ L \mid \text{that can be recognized in time } \mathcal{O}(1) \text{ or } \mathcal{O}(1) \cdot n^{\mathcal{O}(1)} \right\} \]
We say $L$ can be recognized in

\[
\begin{cases}
\text{linear time} & \text{if } L \in \text{TIME}(n) \\
\text{quadratic time} & n^2 \\
\text{cubic time} & n^3
\end{cases}
\]

E.g. If $M$ does not halt on input $S = \text{abba}$, but it halts on every other input, then there is an $M'$ that runs in the same time as $M$, within $O(1)$, that
recognizes the same language and is a decider.

Same for any finite set.

Say that we want an algorithm that on any input $\Sigma^* \cup \text{ASCII}$, reaches $q_{\text{accept}}$ if $P = \text{NP}$, $q_{\text{reject}}$ if $P \neq \text{NP}$.

This language, $L$, is either $\emptyset$ or $\Sigma^* \cup \text{ASCII}$, but either way, this
The language is in time $O(1)$.

If $L = \emptyset$, there is a time 1 alg.

$L = \sum_{i \in \text{ASCII}}$

Linear time, time $O(|L|)$

includes

time $37n+2$,
$16n + 100n^{1/4}$,
$5n + 32 \log(n) + 3$.
Notion

$O(n)$

[due to Udi Manber]

“wh-fh” of n

includes

$log_{10}^{10^{10}}\ n + 3$

\leq Cn$

$O(n)$ can hide huge constants
Example:

\[ \text{CONNECTED\_GRAPH} = \{ \langle G \rangle \mid G \text{ is a graph that is connected} \} \]

\[ \text{connected} = \text{any 2 vertices can be connected by a path} \]

\[ \langle G \rangle = \text{the description of G, we need convolubus} \]
Graph

Vertices

Edges

Insist: \( V = \{1, \ldots, n\} \)
E is a part of V

e.g., \( V = \{1, \ldots, 5\} \)

\[ E = \{\{1,2\}, \{1,3\}, \{1,4\}, \{5,3\}\} \]

Rem: \( \{\ldots\ldots, \{3,5\}\} \)

\( \{\ldots\ldots, \{3,5\}, \{4,1\}\} \)

It same

It same