CPSC 421/501 Nov 6, 2023

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

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

  N = non-deterministic

- P contains all poly time algorithms in CPSC 320

- NP includes 3SAT, SAT, 3COLOUR, PARTITION, ...

Break: Monday 11/13 – 11/15
Monday Wednesday

HW 8 due Nov 16

Indiv 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~~

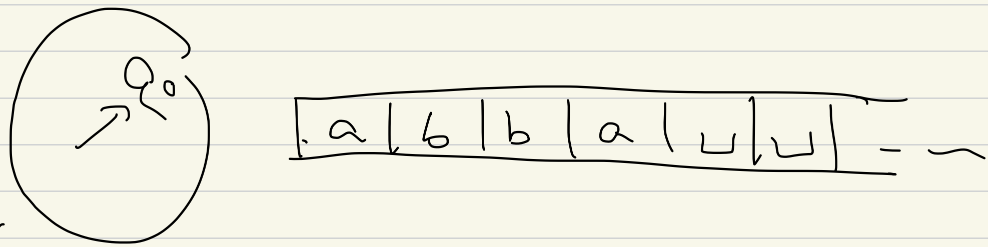
$$P = \{\text{Languages recognizable with polynomial time on a Turing machine}\}$$

$$\left(\text{Remark: } P_{T.M.} = P_{\text{Python}} \right)$$

Warning! Our definitions will slightly vary from [Sip]

If M is a Turing machine:

Initial Config (Step 1)



2nd configuration (Step 2)

⋮

q_{reject}

steps that M takes on input

s is just the difference

step 1 initial config

$\downarrow$

step 2

$\downarrow$

$\vdots$

$\downarrow$

step 100 q_{accept}/q_{reject}

steps = $100 - 1 = 99$

<u>time</u> = # steps

Polynomial time:

If $f, g : \mathbb{N} \to \mathbb{R}$

and $g(n) > 0$ for <u>n sufficiently large</u>, (i.e $\exists n_0$ s.t. $g(n) > 0$ for all $n \geq n_0$), then we write

$$f(n) = O(g(n))$$

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

if there is a C such that

$|f(n)| \leq C\, g(n)$ for <u>n suff. large</u>

i.e. $\exists n_1$ s.t.

$$f(n) \leq C\, g(n)$$

for all $n \geq n_1$

$=$

If $g(n) = g \quad \mathbb{N} \to \mathbb{R}$ s.t.

$g(n) > 0$ for n sufficiently large

$$\mathrm{TIME}(g) \quad \text{OR}$$

$$\mathrm{TIME}(g(n))$$

$= \{$ 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 $\}$.

$$P = \text{PolyTime}_{TM}$$

$$= \bigcup_{k=1,2,\ldots} \text{TIME}(n^k)$$

Rem:

Could also say

$$P = \left\{ L \text{ that can be recognized in time } \left\{ \begin{array}{c} O(1)\, n^{O(1)} \\ n^{O(1)} \end{array} \right\} \right\}$$

We say L can be recognized in

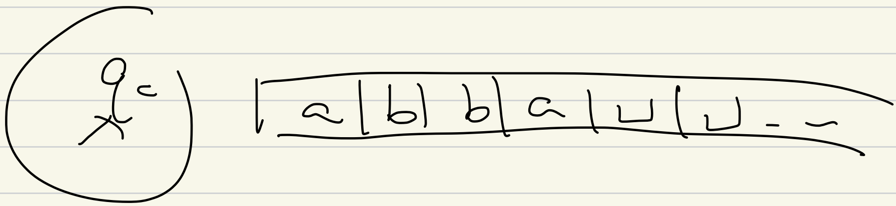
- <u>linear time</u> if $L \in \mathrm{TIME}(n)$
- <u>quadratic time</u> " " " " n^2
- <u>cubic time</u> – – n^3

E.g. If M does not halt on input $s = abba$, 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 Σ^*_{ASCII}, reaches q_{accept} if $P = NP$
" q_{reject} if $P \neq NP$

This language, L, is either $\emptyset$ or Σ^*_{ASCII}, but either way, this language is in time $O(1)$



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

$L = \Sigma^*_{ASCII}$ " " " " " "

Linear time, time $O(n)$

includes

time $37n + 2$,

$16n + 100\, n^{1/4}$,

$5n + 32 \log(n) + 3$ ...

Notion

$$\mathcal{O}\mathcal{O}(n)$$

[due to Udi Manber]

"oh-oh" of n

includes

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

$$\leq C n$$

$O(n)$ can hide huge constants

Example:

CONNECTED_GRAPH

$= \{ \langle G \rangle \mid G$ is a graph that is connected $\}$

connected = any 2 vertices can be connected by a path

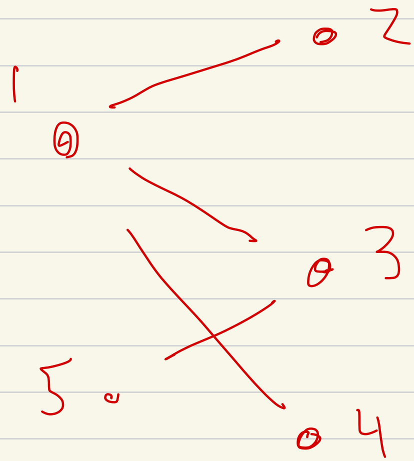
$\langle G \rangle$ = the description of G, we need conventions

Graph

v_2

v_1 vertices

v_3



edges

Insist: $V = \{1, \ldots, n\}$

$E \in$ pairs of V

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

$E = \{\{1,2\}, \{1,3\}, \{1,4\}, \{5,3\}\}$

Rem: $\{$ " " " $\{3,5\}\}$

is same

$\{$ " " $\{3,5\}, \{4,1\}\}$

is same