$\boxed{\mathbb{EXP}}$ & $\mathbb{NEXP}$

$\mathbb{P}$ vs $\mathbb{NP}$ $\qquad\qquad$ $\mathbb{NP}$ vs co-$\mathbb{NP}$

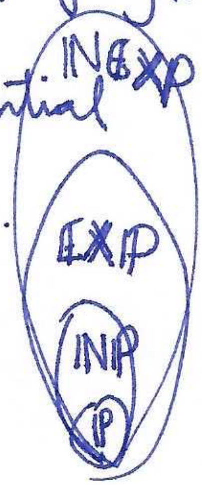
$$\mathbb{EXP} = \bigcup_{c \geq 1} \text{DTIME}(2^{n^c}) \qquad \mathbb{E} = \bigcup_{c \geq 1} \text{DTIME}(2^{cn})$$

$$\mathbb{NEXP} = \bigcup_{c \geq 1} \text{NTIME}(2^{n^c}) \qquad \mathcal{L} \overset{f}{\leq_p} \mathcal{L}'$$

$|x|$ $\quad$ $|f(x)| = \text{poly}(|x|)$

<u>Clm</u>: $\mathbb{NP} \subseteq \mathbb{EXP}$

<u>Proof idea</u>: For $\mathcal{L} \in \mathbb{NP}$, we have an exponential time algorithm that simply tries all certificates.



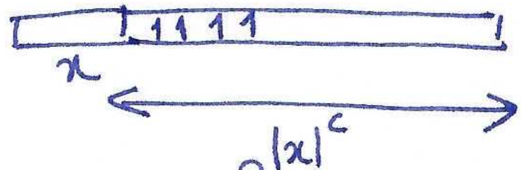
<u>Thm</u>: If $\mathbb{EXP} \neq \mathbb{NEXP}$, then $\mathbb{P} \neq \mathbb{NP}$.

<u>Proof</u>: Padding argument.

Assume $\mathbb{P} = \mathbb{NP}$. Consider $\mathcal{L} \in \mathbb{NEXP}$.

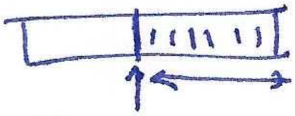
$\exists c \geq 1$, $\mathcal{L} \in \text{NTIME}(2^{n^c})$. There is an NTM M that runs in $O(2^{n^c})$ time and decides $\mathcal{L}$.

$$\mathcal{L}_{pad} = \{ \langle x, 1^{2^{|x|^c}} \rangle \mid x \in \mathcal{L} \}$$



Let us design a NTM M' that decides $\mathcal{L}_{pad}$.

M' (on input w)



1. Checks if $w = \langle x, 1^{2^{|x|^c}} \rangle$. If not, REJECT. $O(n^2)$

2. $\overbrace{\text{Runs } M \text{ on } x}^{\text{simulation}}$, and ~~fo~~ follow output. $O(n)$

Q. What is running time of M'?

✓(A) $O(n^2)$ (B) poly(n) (C) $O(2^{n^c})$

$n = |w|$

$L_{pad} \in NP$. We assume $P = NP$

Therefore, $L_{pad} \in P$. Let us show that $L \in EXP$.

$L_{pad} = \{ \langle x, 1^{2^{|x|^c}} \rangle \mid x \in L \}$

$\exists$ polytime TM N that decides L_{pad}.

Consider N' (on input x)

1. Constructs $\langle x, 1^{2^{|x|^c}} \rangle$.

2. Feeds this to N and follows output.

N' runs in poly(2^{n^c}) time, so $L \in EXP$.

$$P = NP \implies EXP = NEXP$$

$\blacksquare$

Time Hierarchy Theorem

$DTIME(n^c) \subsetneq DTIME(n^{c+1})$

Thm: Let $f, g : \mathbb{N} \to \mathbb{N}$ be time constructible.

such that $\lim_{n\to\infty} \frac{f(n) \log f(n)}{g(n)} \to 0$ $\left(f(n) \log f(n) = o(g(n)) \right)$

$\boxed{\forall n \quad f(n) \log f(n) < g(n)}$

then $DTIME(f(n)) \subsetneq DTIME(g(n))$

Proof: Diagonalization!

~~For any string x~~ Construct machine M

M (on input $\langle N \rangle$) $\quad$ M is like the MOST efficient TM for $L(M)$.

(1) Simulate N on $\langle N \rangle$ (using efficient simulation of Hennie-Stearns) for $g(n)$ steps $\quad n = |\langle N \rangle|$

$\mathcal{U}_{TM}$

(2) If simulation halts, flip output.

Else, reject.

$L(M) \in DTIME(g(n))$

(We will show that $L(M) \notin DTIME(f(n))$).

Suppose, for contradiction's sake, that

$L(M) \in DTIME(f(n))$, decided by TM M'.

M' (on input x) halts in $f(|x|)$ time

Consider some M'' that is equivalent to M' but has large enough encoding length to ensure that ~~$f(x$~~ $f(|\langle M''\rangle|)\log(|\langle M''\rangle|) < g(|\langle M''\rangle|)$

M ~~is equivalent to~~ M'.

$\boxed{L(M) = L(M')}$

runs in $g(n)$ runs in $f(n)$

M' is equivalent to M''.

$L(M'') = L(M')$
$= L(M)$

Run $M(\underbrace{\langle M''\rangle}_{n})$ and see what happens.

$M''(\langle M''\rangle)$ runs in $f(|\langle M''\rangle|)$ time

The simulation (by HS) runs in $f(|\langle M''\rangle|)\log(\ldots)$
$< g(n)$.

So the simulation halts.

Output of $M(\langle M''\rangle)$ is the opposite of $M''(\langle M''\rangle)$.

Contradiction! So $L(M) \notin DTIME(f(n))$. ∎

Thm: $\mathbb{P} \neq \mathbb{EXP}$

$$\bigcup_{c \geq 1} \mathrm{DTIME}(n^c)$$

$$g(n) = 2^n$$

$$\forall c \quad \mathrm{DTIME}(n^c) \subsetneq \mathrm{DTIME}(2^n)$$

$$\not\Rightarrow \bigcup_{c \geq 1} \mathrm{DTIME}(n^c) \subsetneq \mathrm{DTIME}(2^n)$$

$$\mathbb{P} \subseteq \mathbb{NP} \subseteq \mathbb{EXP}$$

$$\mathbb{P} \neq \mathbb{EXP}$$

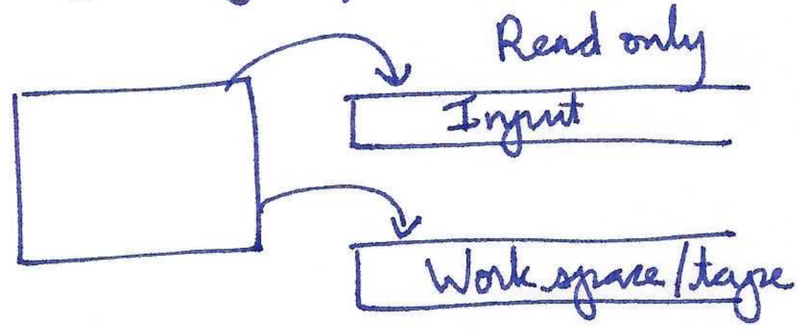
so ~~either~~ $\mathbb{P} \neq \mathbb{NP}$ or $\mathbb{NP} \neq \mathbb{EXP}$.

(We think both.)

Space complexity

To define space complexity, we will use more tapes

(Input tape : read only
work tape : usual
Output tape : final result)



Def: Let $s : \mathbb{N} \to \mathbb{N}$ be space-constructible.

$\text{SPACE}(s(n))$ is the class of languages decided by TMs using at most $c \cdot s(n)$ work space.

constant

$s(n)$ can be less than n !

$$\mathbb{PSPACE} = \bigcup_{c \geq 1} \text{SPACE}(n^c)$$

$$\mathbb{NPSPACE} = \bigcup_{c \geq 1} \text{NSPACE}(n^c)$$

$\mathbb{L} = \text{SPACE}(\log n)$ $\quad$ $\mathbb{NL} = \text{NSPACE}(\log n)$

You need $O(\log n)$ space just to write down the input length.

In $< \log n$ space, it's hard to even decide very simple language.

Configuration Graph

$n = |x|$

Given TM M and input x, we ~~f~~ define
(space bounded) $S(n)$ configuration graph $G_{M,x}$

Recall that the $\boxed{\text{total}}$ $\boxed{\text{state}}$ of M can be represented as a string with tape contents and the state machine state at the head position.

$$\sigma_1 \sigma_2 \sigma_3 - - \; q \sigma_i \sigma_{i+1} - - \; \sigma_{S(n)}$$

$G_{M,x} = (V, E)$ $\quad$ V = set of all such strings of length $\leq S(n)$

= set of all possible configurations

$E = (u, v)$ if M can move from config. u to config v
directed

C_{start} Start config: initial config. of M on input x

C_{Acc} Accepting config: wlog, assume that TM clears out the tape on acceptance, so there is a unique accepting config.

Clm: M accepts x iff $G_{M,x}$ has a directed path from C_{Start} to C_{Acc}.

> Q. For $s(n)$ space (N) TM M, $G_{M,x}$ has
> (A) poly($s(n)$) vertices (B) $O(2^{s(n)})$ ~~(C)~~ $2^{O(s(n))}$

Config. is a string of length $s(n)$ over alphabet $\Sigma \cup Q$ $\#\text{config} = |\Sigma \cup Q|^{s(n)} = 2^{O(s(n))}$

Clm: $G_{M,x}$ has outdegree 1 iff M is deterministic (on x).

Thm: $\text{DTIME}(s(n)) \underset{\text{Easy}}{\subseteq} \text{SPACE}(s(n)) \overset{\text{Trivial}}{\subseteq} \text{NSPACE}(s(n)) \subseteq \text{DTIME}(2^{O(s(n))})$

$(s(n) \geq \log n)$

Proof: Consider $L \in \text{NSPACE}(s(n))$. There is an NTM M using $O(s(n))$ space deciding L.

Design M′ that: constructs the config graph of M on input x.

(1) List out vertices ($2^{O(s(n))}$ time)

(2) List out edges ($2^{O(s(n))}$ time)

(3) Find C_{start}, C_{Acc} ($2^{O(s(n))}$ time)

(4) Run DFS from C_{start} to get path to C_{Acc}.

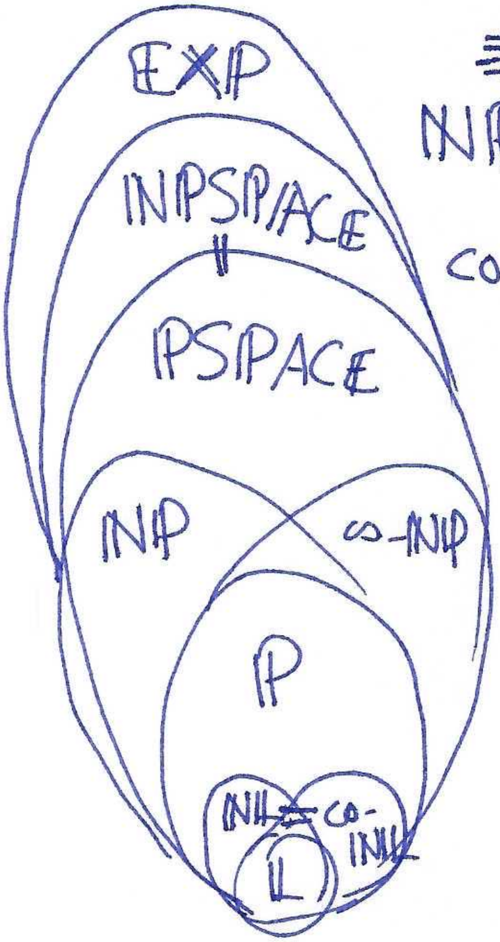
(5) Accept iff path exists.

Clm: $\mathbb{NP} \subseteq \mathbb{PSPACE}$

(try all certificates. Reuse space!)

$\text{co-}\mathbb{NP} \subseteq \mathbb{PSPACE}$

$$\mathbb{L} \subseteq \begin{matrix}\mathbb{NL} \\ \parallel \\ \text{co-}\mathbb{NL}\end{matrix} \subseteq \mathbb{P} \subseteq \mathbb{NP} \subseteq \begin{matrix}\mathbb{PSPACE} \\ \parallel \\ \mathbb{NPSPACE} \\ \parallel \\ \text{co-}\mathbb{NPSPACE}\end{matrix} \subseteq \mathbb{EXP}$$



$\mathbb{P} \neq \mathbb{EXP}$

[Space hierarchy thm]

$\mathbb{L} \neq \mathbb{PSPACE}$

Thm: [Savitch's Theorem 70]

$$\text{NSPACE}(s(n)) \subseteq \text{SPACE}(s(n)^2)$$

(Hence $\bigcup_{c \geq 1} \text{NSPACE}(n^c) \subseteq \bigcup_{c \geq 1} \text{SPACE}(n^{2c})$

$\mathbb{PSPACE} = \mathbb{NPSPACE} = \bigcup_{c \geq 1} \text{SPACE}(n^c)$)

<u>Proof</u>: Consider $L \in NSPACE(s(n))$.

There is a $s(n)$-space NTM M deciding L.

Given input x, consider $\underline{G := G_{M,x}}$

We need to check if $\underline{C_{start}}$ has a path to $\underline{C_{Acc}}$.

G has $\boxed{2^{O(s(n))}}$ vertices. We cannot afford to "write" or construct G completely.

$2^{\alpha s(n)} = K$

Define a procedure $\text{REACH}(C, C', i)$ that outputs (1) True if $\exists$ path from C to C' of length $\leq 2^i$

(2) False otherwise

$\text{REACH}(C_{start}, C_{Acc}, \alpha s(n))$ is True iff $x \in L$.

constant

(Base case) ~~REACH(Cstart, C~~

$\underleftrightarrow{\text{REACH}(C, C', 0)}$ is True iff $(C, C') \in E$

TM M can go to config C' from C

$\text{REACH}(C, C', 0)$

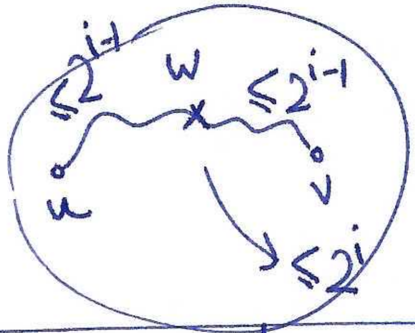
can be decided in $O(\alpha s(n))$ space.

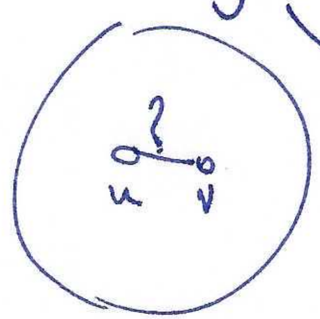
Consider directed graph $G=(V,E)$. $|V|=k=2^{O(s(n))}$

$$\text{REACH}(u,v,i) = \exists w \in V\ \big(\text{REACH}(u,w,i-1) \wedge \text{REACH}(w,v,i-1)\big)$$

REACH$(u,v,0)$ can be decided in $O(\log k)$ space

$s(n)$





REACH(u,v,i) :

$(u,v) \in E$

(1) If $i=0$, decide REACH$(u,v,0)$ in $O(\log k)$ space.

(2) For all vertices w: (Reuse space!)

(a) ~~Check if~~ Run REACH$(u,w,i-1)$ ~~is True~~

(b) Run REACH$(w,v,i-1)$

(c) If both are True, output True.

(3) Output False

<u>Clm</u>: REACH(u,v,i) outputs True iff there is a path of length $\leq 2^i$ from u to v.

Proof: $\boxed{\text{Induction on } i}$ Exercise.

How much space to implement/run REACH(u,v,i)?

<u>$sp(i)$</u> = space complexity of $\max_{u,v}$ REACH(u,v,i)

What is the recurrence for $s(i)$?

$$s(0) = O(\log K)$$

✓ (A) $s(i+1) \leq s(i) + O(\log K)$

(B) $s(i+1) \leq 2s(i) + O(\log K)$

(C) $s(i+1) \leq K\, s(i) + O(\log K)$

First store w. $(O(\log K))$ space.

→ Compute REACH$(u, w, i-1)$ $s(i)$ space.

Store the answer. $O(\log K)$ space

Compute REACH$(w, v, i-1)$ Reuse space

Store the answer $O(\log K)$ space

Check both answer. If true, done.

Otherwise, increment w, clear all other space

Thus, $s(i) = O(i \log K)$

$$s(\log K) = O(\log^2 K)$$

The path length is at most $K = 2^{\log K}$.

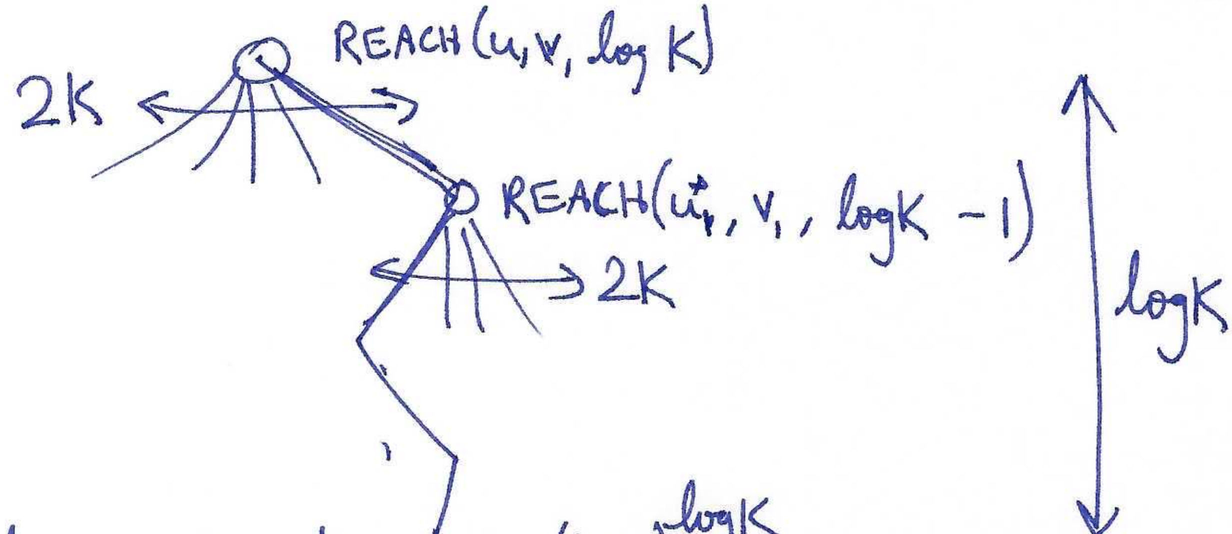
So to determine if there is a path of any length from u to v ~~requires~~ can be done in $O(\log^2 K)$ space.

deterministic

For Savitch's theorem, $K = 2^{\alpha s(n)}$

$$O(\log^2 K) = O(s(n)^2) \text{ space}$$

We can decide $L \in NSPACE(s(n))$ in deterministic $O(s(n)^2)$ space. ∎



$$\text{Size of recursion tree} = (2K)^{\log K}$$
$$= 2^{O(\log^2 K)}$$

$$\text{Size of recursion stack} = O(\log^2 K)$$