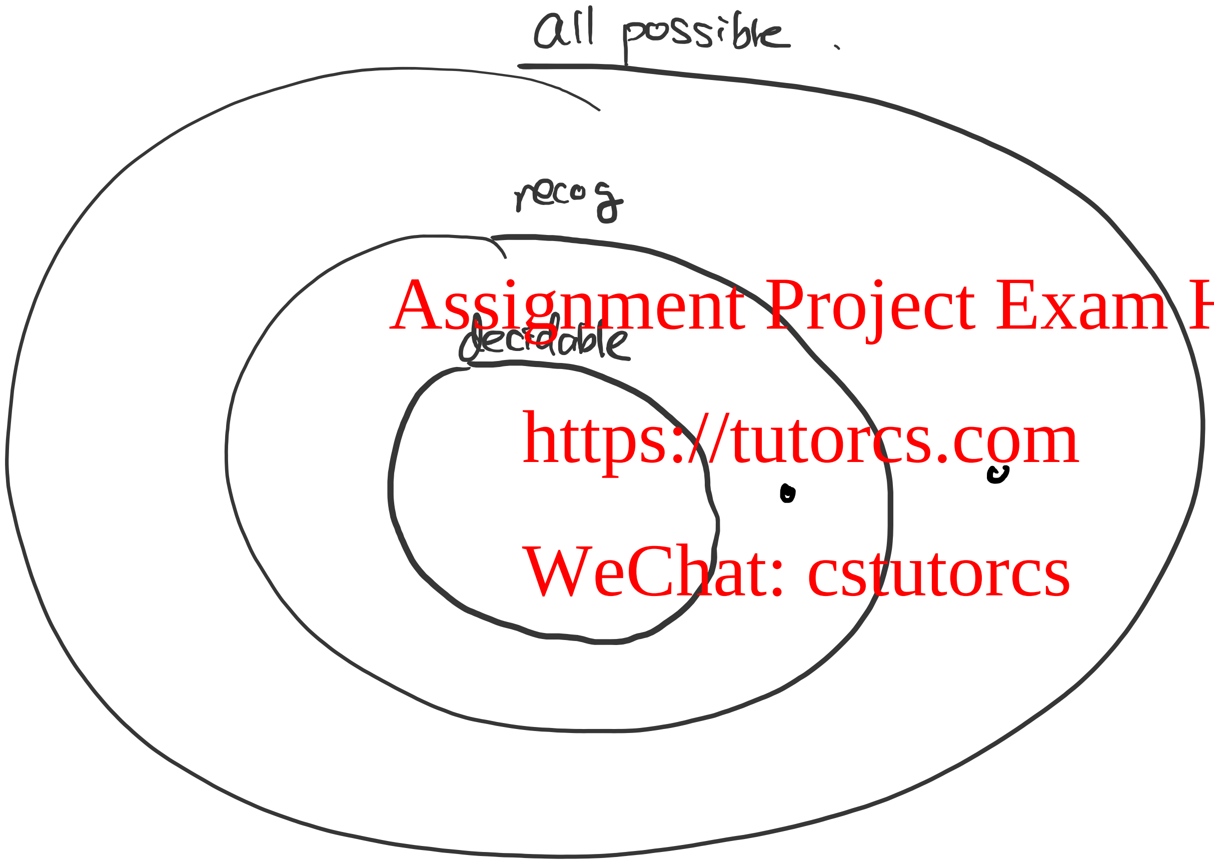
Lecture 11

What have we covered … ?



Infinite Resource

Not a realistic assumption.

finite space & time for practical purposes.

Reality : Limited **Time** and **Space**

Model this.

Machine independent way.

asymptotic notation
as a function of
length of the input.

Turing Machine Limitations ?

Limit # of steps (asymptotically).

Limit size/length of the tape T.M. uses.

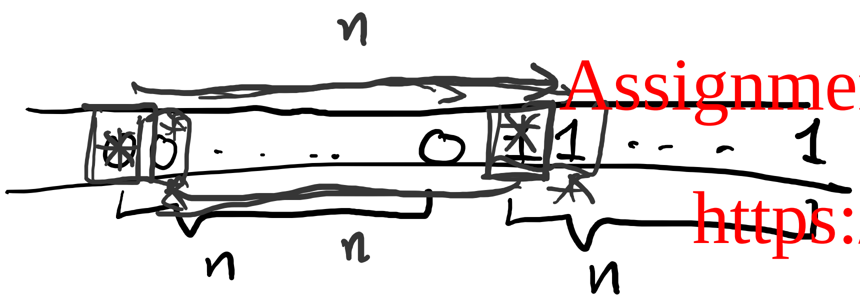
Circuit Limitations ?

Time Complexity Class

- Let's focus on time first – but how do we measure time ?

(ex) 0^n 1^n Runtime

$\mathcal{L} := \{0^n 1^n \mid n \geq 0\}$. is decidable by a D.T.M (single tape) in $O(n^2)$ time.



accept if nothing remains

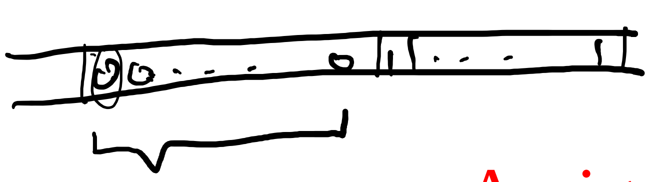
o/w reject.

$$n * (2n) = O(n^2)$$

ex. Can we decide this $\mathcal{L}$ in $O(n^{1.5})$ time?

Is this the best bound ?

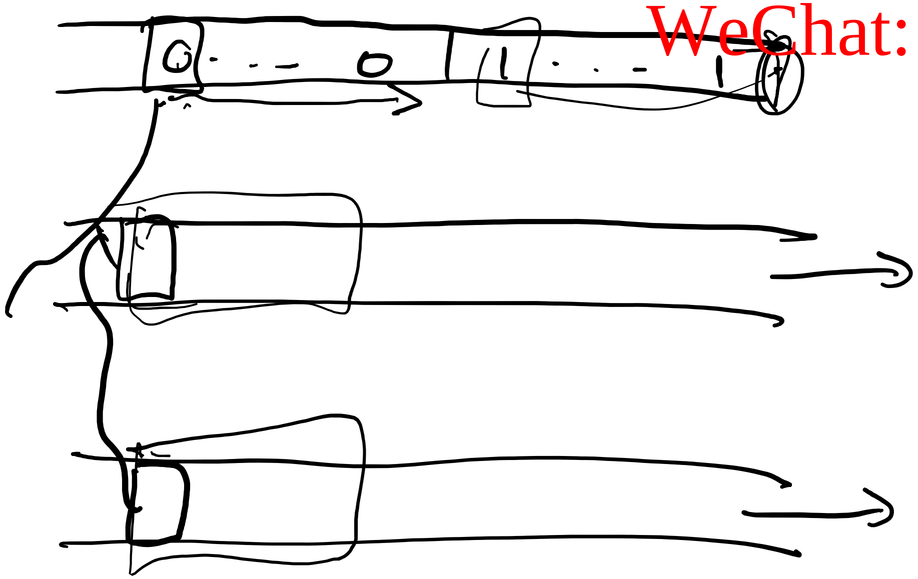
in random access word-RAM computer



increment the counter for 0

increment the counter for 1

this is $O(n)$

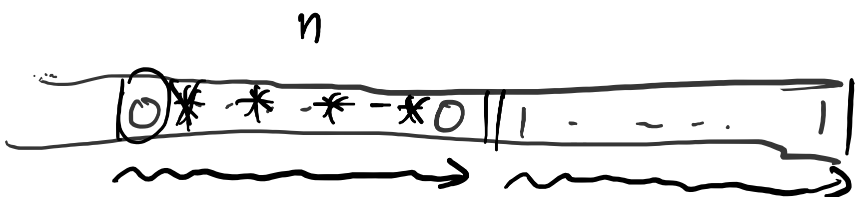


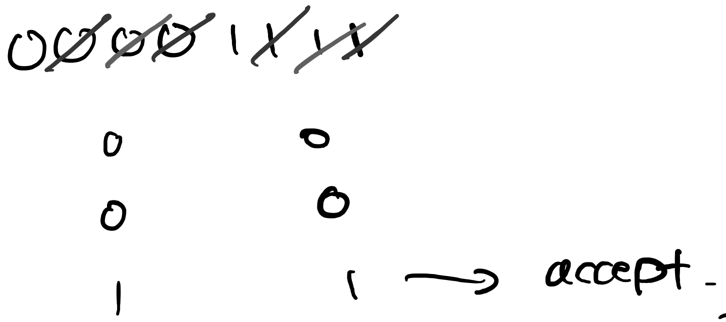
store counter for 0

1

$O(n)$ time.

Better Algorithm (half)





Compute the parity at each side. $O(n)$ steps

if not equal, reject

if equal, delete consecutive 0, consecutive 1 → $O(n)$ - step. → $O(n)$ → $O(\log n)$

$\lceil \frac{n}{2} \rceil$

check parity again.

accept if one zero ~~and~~ one one remains at the end.

o/w reject.

Total # of steps

$O(n \log n)$

Definition DTIME$(t(n))$

n^2

$L \in DTIME(t(n))$ if L can be decided by a D.T.M. (single tape) in $O(t(n))$ time (worst case)

for ex $L_{eg} \in DTIME(n \log n) \subseteq DTIME(n^2)$

previous.

More generally $t_1(n) \ll t_2(n)$

$DTIME(t_1(n)) \subseteq DTIME(t_2(n))$ (by definition)

But what if use other models ?

- Unlike decidability, time complexity is sensitive to model used

Multi-tape Simulation

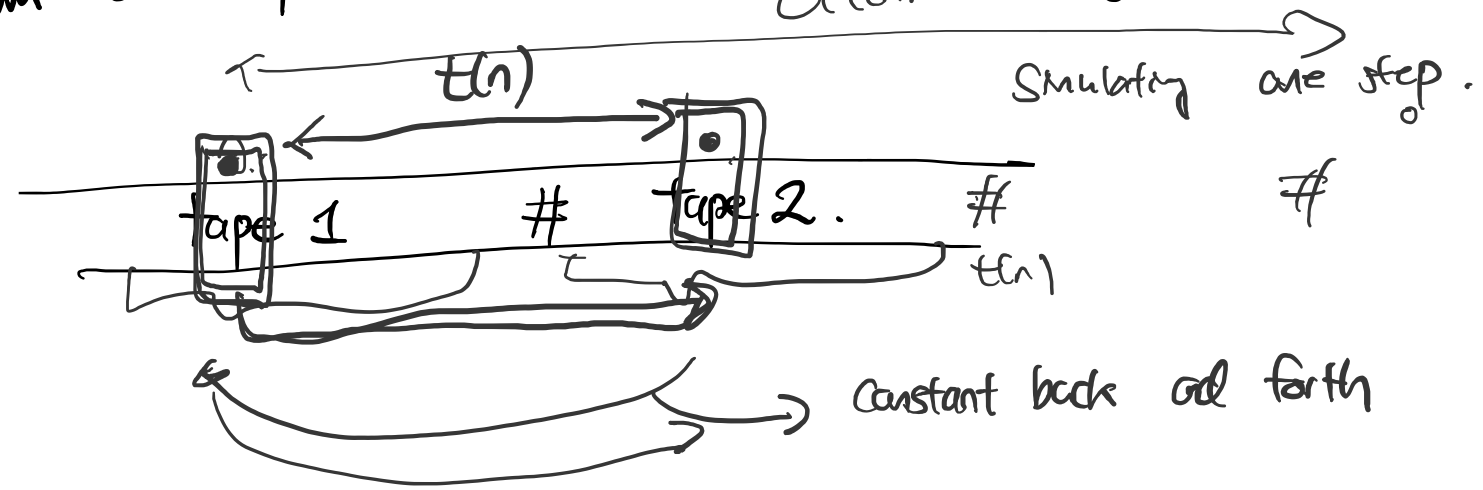
$t(n) \geq n$

- $\boxed{\text{Each step can be simulated in } O(t(n)) \text{ time – why ?}}$
- So total runtime ?

If L can be decided by a Multi ~~two~~ tape D.T.M M in $\underline{t(n)}$ time, then

L " by a single tape $O(t(n)^2)$ time. Quadratic overhead.

proof from our previous proof of simulating $O(t(n))$ Multitape by single tape.



Nondeterministic TM simulation

L can be decided by NTM in $O(t(n))$ if

$x \in L$, N accepts x in time $O(t(n))$

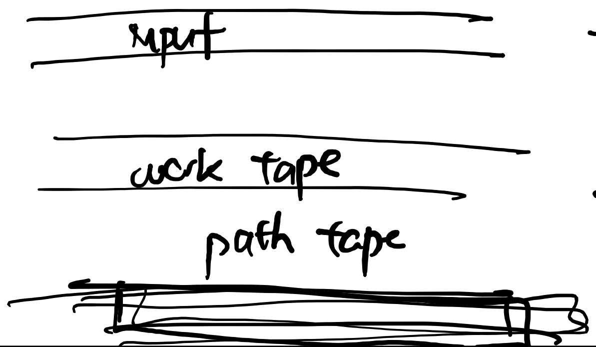
$x \notin L$ N ~~doesn't accept~~ rejects x in t

$t(n) \cdot 2^{O(t(n))}$

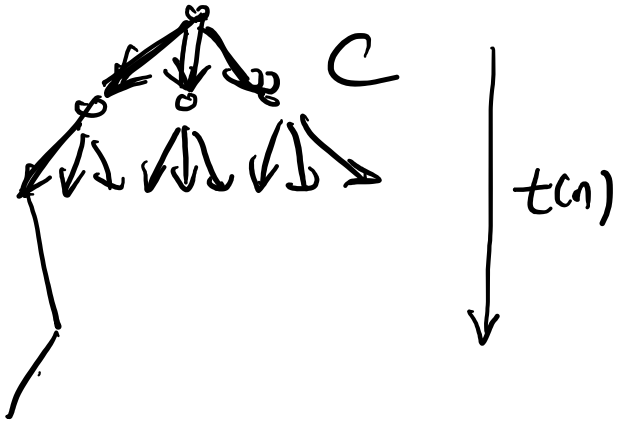
$= 2^{O(t(n))}$

Claim) If N runs in $t(n)$-time, can be simulated by Multi-tape T.M. $2^{O(t(n))}$-time.

pf) We have seen it!!



$t(n)$ time.



$C^{t(n)}$

$= (2^{\log C})^{t(n)}$

$= 2^{O(t(n))}$

Efficient Algorithm

- Polynomial Time Algorithms !

Class P $:= \bigcup_{k \in \mathbb{N}} TIME(n^k)$

- Captures any language that can be decided in polytime (i.e. reasonable amount of time) → does not matter if we use single-tape vs. Multi-tape.
- Ex) PATH

Strong Church Turing Thesis (debatable)

Due to Quantum Computer.

- Any Language that can be decided in (Polynomial Time) Computing Machine can be decided in (Poly Time) via DTM

single tape.

Class NP

- Poly time via NTM

NTIME($t(n)$) : L can be decided by N.T.M in $O(t(n))$-time.

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$$

↪ if $L \in NTIME(\underbrace{t(n)}_{\text{poly}})$, then $L \in DTIME(\underbrace{2^{O(t(n))}}_{\text{no longer polynomial}})$.