# COL 332 Lecture 35 April 20

- A language L is in DTIME$(f(n))$
  if there is a det TM M with
  time complexity $f(n)$ $L(M) = L$

- A language L is in NTIME$(f(n))$
  if there is a non-det TM with
  time complexity $f(n)$

- A language is in DSPACE$(f(n))$
  . . . . - - - det TM with space compl $f(n)$

- A language L . . . . NSPACE$(f(n))$
  if there is NDTM with space complexity $f(n)$

Complexity-theory is about exploring general relations between the complexity classes DTIME, NTIME, DSPACE, NSPACE

For example, if $L$ is in DTIME $(f(n))$ then $L$ is in DSPACE $(f(n))$

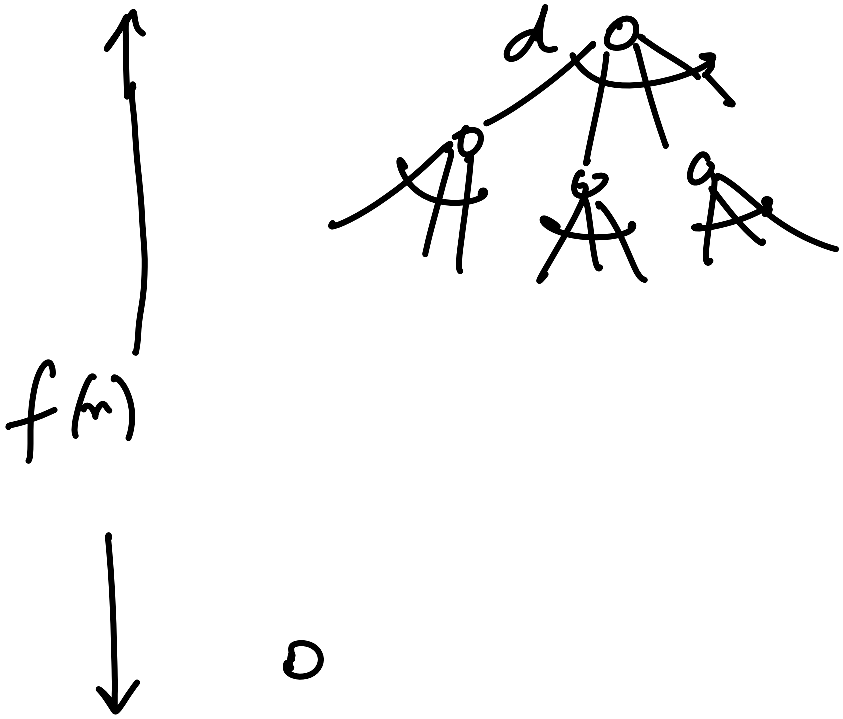
Relation between DTIME( ) and NTIME( )

If $L \in \mathrm{DTIME}(f(n))$

$\Rightarrow L \in \mathrm{NTIME}(f(n))$

?? If $L \in \mathrm{NTIME}(f(n))$ what does it imply for DTIME( )

# NDTM simulated by a DTM




The cost of simulation is the # nodes in the tree

$$O\left(d^{f(n)}\right)$$

$$2^{O(\log d)\cdot f(n)}$$

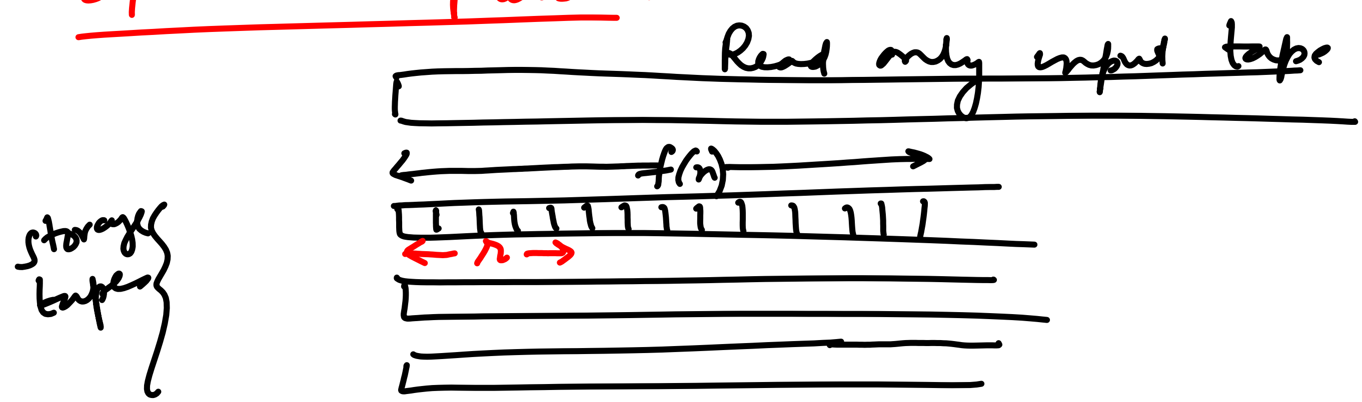
For $f(n)$ being a polynomial function, i.e. $n^c$ for some fixed $c$, we are confronted with the $P = NP$ problem

Claim: If $L$ is in the class DSPACE$(f(n))$ then $L$ is also in the class

DSPACE$(c f(n))$

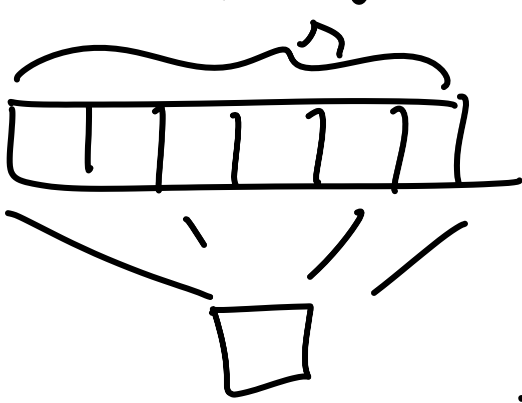
for any $c > 0$

## Space compression




We want to reduce the space by a factor $r \sim \frac{1}{c}$ (integer)

All $r$ tuples in the storage tape are compressed into a new symbol




We must keep track of which of the $r$ cells the head is positioned on.

This can be done by increasing the # states, so that once it is outside a window of $r$ cells, the new machine must actually move the head.

**Claim**: If $L \in \mathrm{DTIME}(f(n))$ then

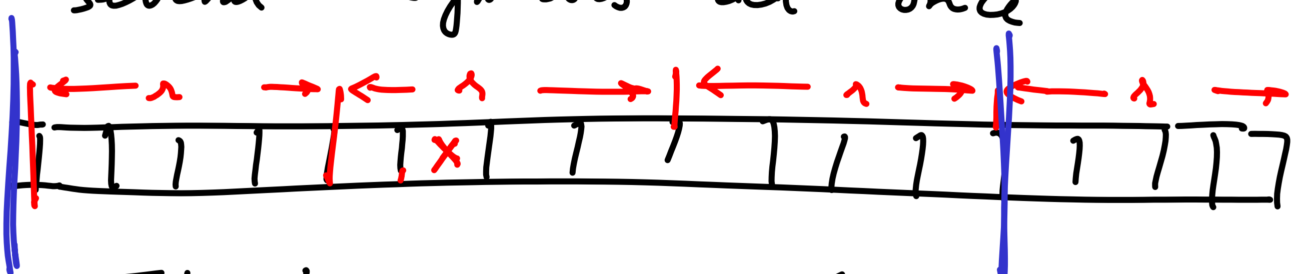
$$L \in \mathrm{DTIME}(c\, f(n))$$

if $T(n)$ is $\omega(n)$

$$\left[ \lim_{n\to\infty} \frac{T(n)}{n} \to \infty \right)$$

## Time Compression

Will involve compressing space so that the new machine can read several symbols at once




The head will definitely stay within the blue region

To simulate $r$ steps of the old machine, we somehow must precompute the $r$-fold composition of the transition function $\delta$ : $\delta^r$

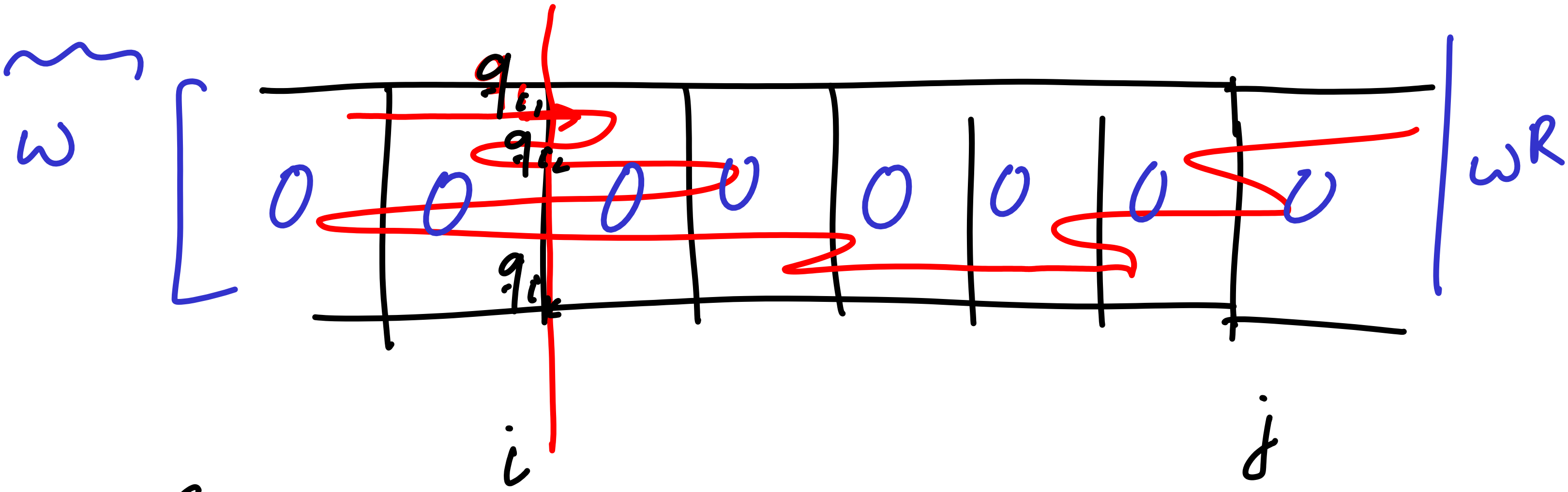
For the language $L_{Pal} = \{ w c w^R \}$

What is the time complexity in a 1 tape TM.

(In a 2 tape TM, $L_{pal} \in n$)

Consider all strings of the form $\Sigma^{n/4} \, 0^{n/2} \, \Sigma^{n/4}$ which are palindromes

$$\underbrace{a\,b\,c}_{n/4}\;\underbrace{0\,0\,0\,0\,0\,0\,0}_{n/2}\;\underbrace{c\,b\,a}_{n/4}$$




Crossing sequences : is a order set of states on a cell boundary

$\mathcal{C}_i$ : the crossing sequence in $i^{th}$ boundary

$\mathcal{C}_j$ : " " " " $j^{th}$ boundary

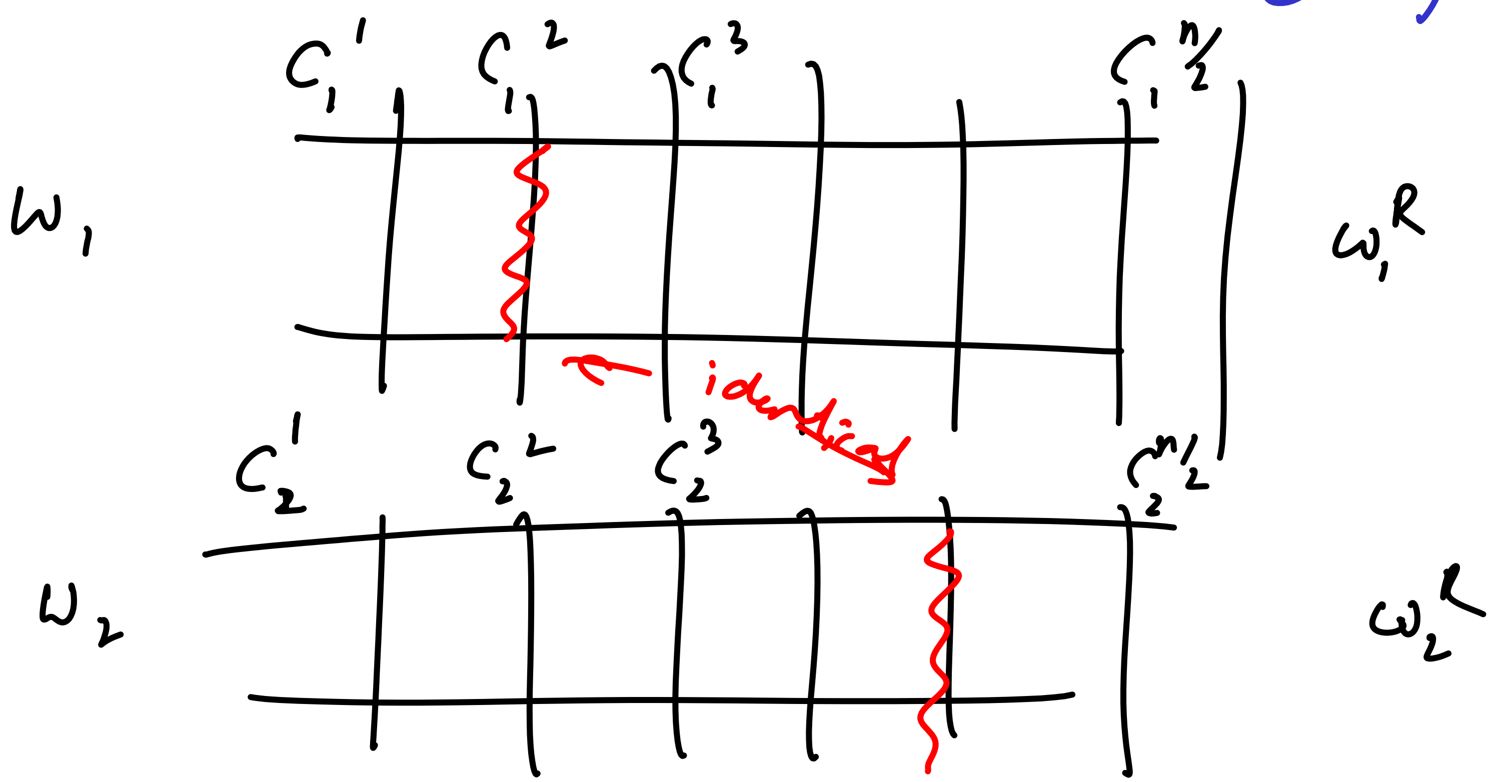
We want to prove the following property of the crossing sequences for any TM that correctly recognises palindromes of the form $\Sigma^{n/4} 0^{n/2} \Sigma^{n/4}$

Claim: For at least one input, the TM has the min length crossing sequence on the portion $0^{n/2}$ as $\Omega(n)$

Observation: For any two distinct palindromes of the form

$$w_1 \; 0^{n/2} \; w_1^R$$
$$w_2 \; 0^{n/2} \; w_2^R \qquad w_1 \neq w_2$$

no two crossing sequences can be identical (in the window $0^{n/2}$)




Then the TM can be fooled into accepting $w_1 \, 0^k \, w_2^R \quad w_1 \neq w_2$

How many possible $w_1$'s of length $n/4$

For binary alphabet $2^{n/4}$

So there must be at least $2^{n/4}$ distinct crossing sequences

If the min length crossing sequence (min over all possible inputs) for a fixed TM is $K$. Then the number of crossing sequences of length at most $K$ is $\sim |Q|^K$

$$|Q|^K \geq 2^{n/4}$$

$$\Rightarrow \quad K \geq \Omega\left(\frac{n}{\log|Q|}\right)$$