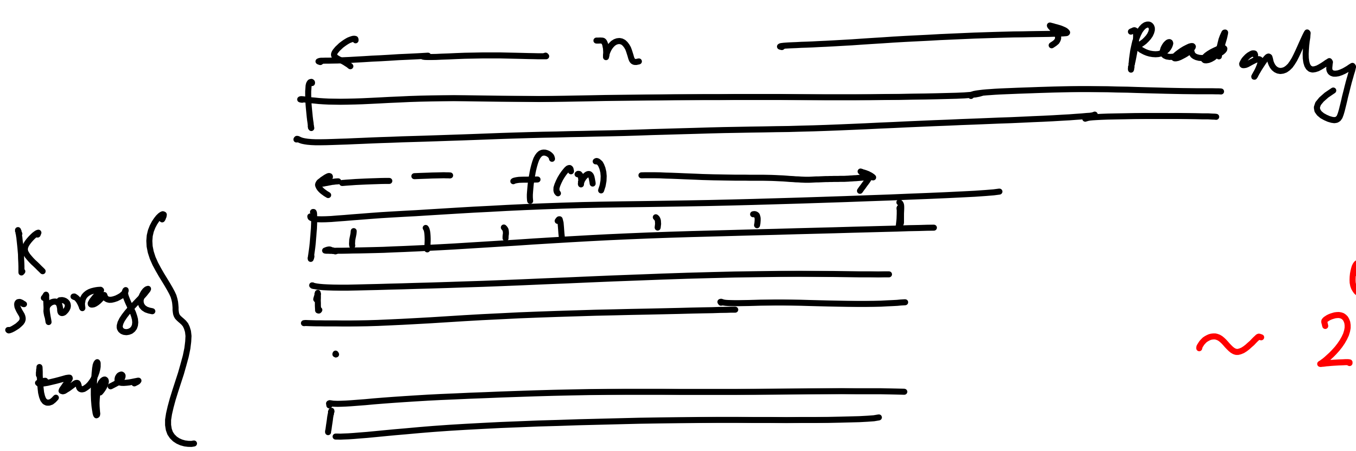
COL 352 Lecture 36 April 23

	DTIME	NTIME	DSPACE	NSPACE
DTIME(f(n))	X	f(n)	f(n)	
NTIME(f(n))	$2^{O(f(n))}$	X		
DSPACE(f(n))	$\boxed{2^{O(f(n))}\ \ f(n) \geq \log n}$		X	
NSPACE(f(n))			$\boxed{(f(n))^2}$ SAVITCH'S THM	X



$\sim 2^{O(f(n))}$

Count the maxm # of distinct IDs

ID: # states, tape contents, head positions

$$\leq |Q| \times |\Gamma|^{kf(n)} \times f(n)^k \times \boxed{n}$$

$$P = \bigcup_{i \geq 1} \text{DTIME}(n^i)$$

$$NP = \bigcup_{i \geq 1} \text{NTIME}(n^i)$$

$$\text{PSPACE} = \bigcup_{i \geq 1} \text{DSPACE}(n^i)$$

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

Union then these are well defined and consistent

$$P \stackrel{?}{=} NP$$

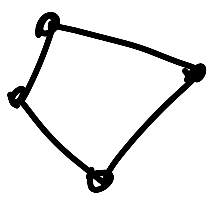
By our previous observation if there is a NDTM in $\text{NTIME}(n^2) \Rightarrow$ there is a DTM in $\text{DTIME}(2^{n^2})$

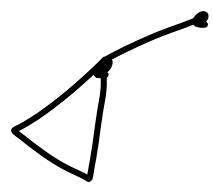
$n^{\log n}$ is $\omega(n^{1000})$

For many "daily life" problems we are able design non-det polynomial time algorithms but are unable to design deterministic algorithms.

<u>Eg</u> Hamiltonian Cycle problem

Given a graph, find a tour that visits every vertex exactly once.





<u>Note</u>: TSP is an optimization problem

Given a weighted graph produce the hamiltonian cycle of least weight.

Suppose I claim that I have a soln to the H.C. problem

The soln is "efficiently" verifiable (polynomial-time)

The process of verification is a non-deterministic algorithm.

1. Guess (non-det) a cycle
2. Verify in polynomial time

Many of the NP problems are actually "equivalent" in hardness

NP - completeness (under polynomial time reductions)