Chapter 6. Time Complexity.

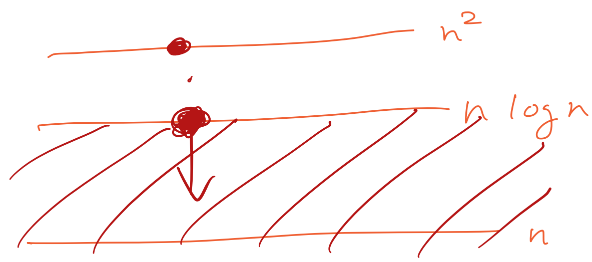
To get the time complexity of an algorithm that solves a problem (recognizes a language), we analyze the algorithm. Say $O(2^n)$

What does that say about the inherent hardness of the problem (language)? That it is "no harder" than $O(2^n)$ to solve.

... but maybe there is an n^5 algorithm, we just haven't found it yet.

We want to explore the complexity or hardness of the problems themselves. An alg that decides it just gives an upper bound.

Eg sort(n)



Defn 7.1 Let M be a det-TM that halts on all input
The running time or time complexity of M is a function $f: \mathbb{N} \to \mathbb{N}$
where $f(n)$ is the max # of steps that M takes on inputs of size n. Customarily n is the size of the input.
We call M a $O(f(n))$-TM

Defn 7.2 f, g functions, $f, g: \mathbb{N} \to \mathbb{N}$, $f(n) \in O(g(n))$
if $\exists$ positive numbers c, n_0 such that
$f(n) \leq c \cdot g(n) \ \forall n \geq n_0$.
$g(n)$ is called an asymptotic upper bound for $f(n)$.

Is O transitive? $f(n) \in O(g(n)) \in O(h(n)) \Rightarrow f(n) \in O(h(n))$.

Defn 7.5 f, g functions $f, g : \mathbb{N} \to \mathbb{R}^+$

$f(n) \in o(g(n))$ iff $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

Defn 7.7 $\text{TIME}(t(n))$ = collection of all languages decidable by a $O(t(n))$ - TM.

Eg $A = \{ 0^k 1^k \mid k \geq 0 \}$.

0	0	0	0	1	1	1	1	␣	␣	␣	␣

Clearly A is in $\text{TIME}(n^2)$

Alg 1 "1. Scan and reject if $\exists$ 0's after 1's. $\}$ $O(n)$

2. Repeat while $\exists$ both 0 and 1 on tape: $\}$ $O(n^2)$

   2.1 X-out leftmost 0, leftmost 1 $\}$ $O(n)$

3. If there are 0's or 1's left, REJECT. $\}$ $O(n)$

   If there are neither, ACCEPT."

Running time is $O(n^2)$

ie A does not require more than $c \cdot n^2$ time to recognize... but can it be done in less time?

Alg 2 : "1. Scan L to R checking that no 0 follows a 1.

2. If $\nexists$ 0's and $\nexists$ 1's on tape, ACCEPT

   2.1 Check that

   parity of $\#_0(\text{tape})$ = parity of $\#_1(\text{tape})$

   as you would using a DFA;

   if not equal parity, REJECT.

2.2 X-out every 2^{nd} 0
X-out every 2^{nd} 1.
Go to 2"

Alg 2 makes $O(\lg n)$ passes over contents of tape.
Occupied length is $2n$
$\therefore$ running time is $O(n \log n)$.