Introduction to Time Complexity

31 March 2021 00:48

Field of Complexity Theory.

- Computational models like Time, Space, Random, Circuits, Multiparty Computation,
- Dividing the Computational Problems based on the amount of resources necessary.
  - ↳ Complexity Classes.
- In this course we will see mainly two things,
  - Time Complexity & - Space Complexity.

* Only decidable languages from now on.
* We will do a more careful analysis of the resources used.
* In this chapter, it is Time Complexity.

Def. 7.1: Running time or time complexity of a Turing machine M is the function $f: \mathbb{N} \to \mathbb{N}$ where $f(n)$ is the maximum time taken by M to accept / reject an input of length n.

Def 7.2: Landau's O-notation.

$3n^2 + 5n + 2$

$3n^2 + 5n + 1$

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

Def 7.2: Landau's O-notation.

If $f, g : \mathbb{N} \to \mathbb{R}^+$, we say that

$f(n) = O(g(n))$ if $\lim_{n \to \infty} \frac{f(n)}{g(n)} \leq c$ for some const. c.

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

$f(n) = \Omega(g(n))$ if $\lim_{n \to \infty} \frac{f(n)}{g(n)} \geq c$ for some const. c.

$$\left.\begin{array}{c} f(n) = O(g(n)) \\ \text{and} \\ f(n) = \Omega(g(n)) \end{array}\right\} \Rightarrow f(n) = \Theta(g(n)).$$

Exercise: Familiarise yourself with these symbols.

Introduction to Theory of Computation by Michael Sipser.

Def 7.7: DTIME(t(n)) (TIME(t(n)) in the book) is the set of languages that are decided by an $O(t(n))$ time TM.

Deterministic, may be multitape. Our main concern is if $t(n)$ is a polynomial. So n^2 and n^3 are both OK.

$n \log n \approx O(n^2)$

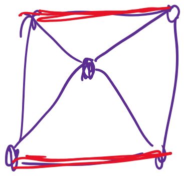
$= O(n^{1.1})$

n^3 are both OK.

$n \log n = O(n^{1.1})$

Def 7.12: $P = \bigcup_{k=1}^{\infty} DTIME(n^k)$.

Class P : Why class P?



- No more machine level description.
- Robust class. Independent of most models of computation.
- Stands for all efficient / practical algorithms
- We need deterministic.
- Exponential vs Polynomial

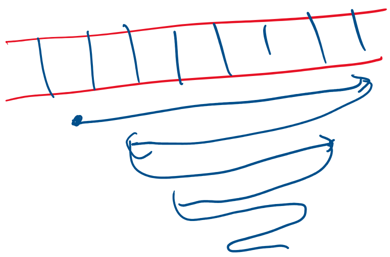
$\downarrow$ Brute force $\qquad\qquad$ $\downarrow$ $n, n^2, n^3, \dots$

2^n *

$|S| = 100$
is there a subset T of S such sum of elements of T is 2000?

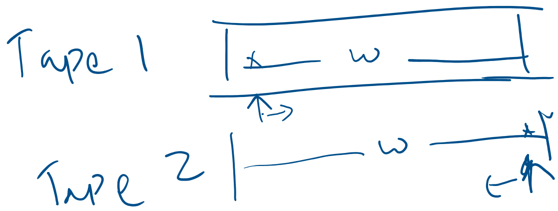
PALINDROME $= \{ w \mid w = w^R \}$

Let $n = |w|$.



1. Check $w_1, w_n, w_2, w_{n-1}, \dots$
2. Accept if for all $i \leq n/2$, $w_i = w_{n+1-i}$.
3. Else reject.

Have to make $(n-1)+(n-2)+(n-3)+\dots = O(n^2)$ moves on the tape.



Algorithm 2

1. Copy w to tape 2.
2. Move head of tape 2 to right most end.
3. As head of tape 1 moves from L to R, head of tape 2 moves from R to L.
4. Accept if $w = w^R$.

In this, tape heads move $O(n)$ steps. But we need 2 tapes.

Theorem 7.8: Let $t(n)$ be such that $t(n) \geq n$. Then every multitape TM running in $t(n)$ time has an equivalent single tape TM running in time $O((t(n))^2)$.

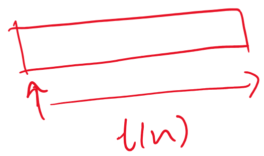
Proof: Let k be the number of tapes. We simulate the k tapes in a single tape as follows.

tape 1 $\hat{0}$	#	tape 2	#	. . .	#	tape k

0. Put the tape in the above format.

1. Make 1 pass for head locations.

2. Simulate the transition of a k-tape machine.



3. Make another pass to update.

The k-tape machine takes t(n) time. Each pass of the single-tape requires $k * t(n)$ time. We need to make $2t(n)$ passes.

Total time need for the single tape machine $\Big\}$ $= 2t(n) * k\, t(n)$

$= O\left((t(n))^2\right)$.

The class P : n^k for some constant k.

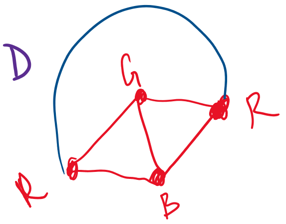
Polynomial : $O(n^k)$

Exponential : $O(k^n)$. → Much more faster growth.

$$P = \bigcup_{k=1}^{\infty} DTIME(n^k).$$

Examples:

(1) CONNECTED



n $\quad 3^n$

(2) 3-COLORABLE

(3) RELPRIME $\quad O(\sqrt{N})$? $\quad n = \log N$

(4) PRIME $\longrightarrow$ N

(5) $A \times B$ for $n \times n$ matrices $\longrightarrow$ $n^{2.87} = n^{\log_2 7}$

(6) FACTORS of a number k.

(7) TSP : Travelling Salesman Problem.

(8) SUBSET-SUM.