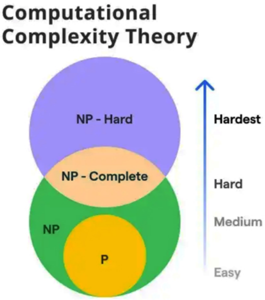
▷ Complexity Theory

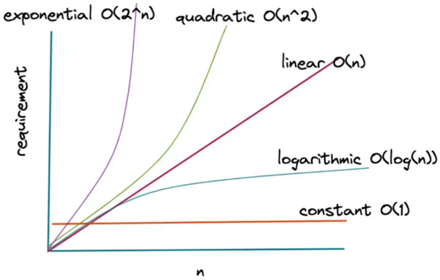
P and NP

P : Problems <u>solvable</u> in polynomial time.

NP: Problems <u>verifiable</u> in polynomial time

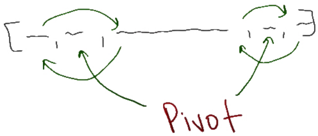
NPC: Hardest problems in NP.





▷ Typical Problems and Algorithms

Quick Sort



8	3	1	7	0	10	2

SAT

Boolean Satisfiability

Boolean Formula
1. Variables $x_1\ x_2\ x_3$ $\left\{ \begin{array}{l} \text{"true"} / 1 \\ \text{"false"} / 0 \end{array} \right.$
2. "not" $\bar{x}_1$ $\bar{x}_2$ $(x_1 = \text{true} \Rightarrow \bar{x}_1 = \text{false})$
3. "and" $(x_1 \wedge x_2)$ $0 \wedge 0 = 0$ $1 \wedge 0 = 0$ $0 \wedge 1 = 0$ $1 \wedge 1 = 1$
4. "or" $(x_1 \vee x_2)$ $0 \vee 0 = 0$ $1 \vee 0 = 1$ $0 \vee 1 = 1$ $1 \vee 1 = 1$

QUIZ. $((x_1 \vee \bar{x}_2) \vee x_3) \wedge (\bar{x}_1 \vee x_2)$ $x_1 = 1$ $x_2 = 1$ $x_3 = 1$

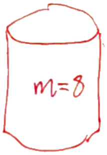
0/1 Knapsack Problem

$m = 8$
$n = 4$

$P = \{1, 2, 5, 6\}$

$w = \{2, 3, 4, 5\}$

$x_i = 0/1$ $x = \{1, 0, 0 \quad \}$



Boolean Equations Solving

$x_1 \cdots x_n \in \mathbb{F}_2$

$$\left\{ \begin{array}{l} x_1 x_2 + x_1 + x_3 + 1 = 0 \\ x_1 x_3 + x_2 x_3 + x_2 + 1 = 0 \\ x_2 x_3 + x_1 + x_2 + x_3 = 0 \end{array} \right.$$

▷ Big O

名称	复杂度类	运行时间（$T(n)$）	运行时间举例	算法举例
常数时间		$O(1)$	10	判断一个二进制数的奇偶
反阿克曼时间		$O(\alpha(n))$		并查集的单个操作的平摊时间
迭代对数时间		$O(\log^* n)$		分散式圓環著色問題
对数对数时间		$O(\log\log n)$		有界优先队列的单个操作[1]
对数时间	DLOGTIME	$O(\log n)$	$\log n$，$\log n^2$	二分搜索
幂对数时间		$(\log n)^{O(1)}$	$(\log n)^2$	
（小于1次）幂时间		$O(n^c)$，其中 $0 < c < 1$	$n^{\frac{1}{2}}$，$n^{\frac{2}{3}}$	K-d树的搜索操作
线性时间		$O(n)$	n	无序数组的搜索
线性迭代对数时间		$O(n \log^* n)$		莱姆德·赛德爾的三角分割多边形算法
线性对数时间		$O(n \log n)$	$n \log n$，$\log n!$	最快的比较排序
二次时间		$O(n^2)$	n^2	冒泡排序、插入排序
三次时间		$O(n^3)$	n^3	矩阵乘法的基本实现，计算部分相关性

名称	复杂度类	运行时间（$T(n)$）	运行时间举例	算法举例
多项式时间	P	$2^{O(\log n)} = n^{O(1)}$	n，$n \log n$，n^{10}	线性规划中的卡馬卡演算法，AKS质数测试
准多项式时间	QP	$2^{(\log n)^{O(1)}}$		关于有向斯坦纳树问题最著名的 $O(\log^2 n)$ 近似算法
次指数时间（第一定义）	SUBEXP	$O(2^{n^\epsilon})$，对任意的 $\varepsilon > 0$	$O(2^{(\log n)^{\log\log n}})$	假設複雜性理論推測，BPP 包含在 SUBEXP 中。[2]
次指数时间（第二定义）		$2^{o(n)}$	$2^{n^{1/3}}$	用於整數分解與圖形同構問題的著名演算法
指数时间	E	$2^{O(n)}$	1.1^n, 10^n	使用动态规划解决旅行推销员问题
阶乘时间		$O(n!)$	$n!$	通过暴力搜索解决旅行推销员问题
指数时间	EXPTIME	$2^{\text{poly}(n)}$	2^n, 2^{n^2}	
双重指数时间	2-EXPTIME	$2^{2^{\text{poly}(n)}}$	2^{2^n}	在預膨脹算術中决定一個給定描述的真實性