

P, NP, NP complete, NP hard

- 参考资料

<https://people.orie.cornell.edu/dpw/orie6300/Lectures/lec25.pdf>

<http://www.doc88.com/p-2778214890655.html>

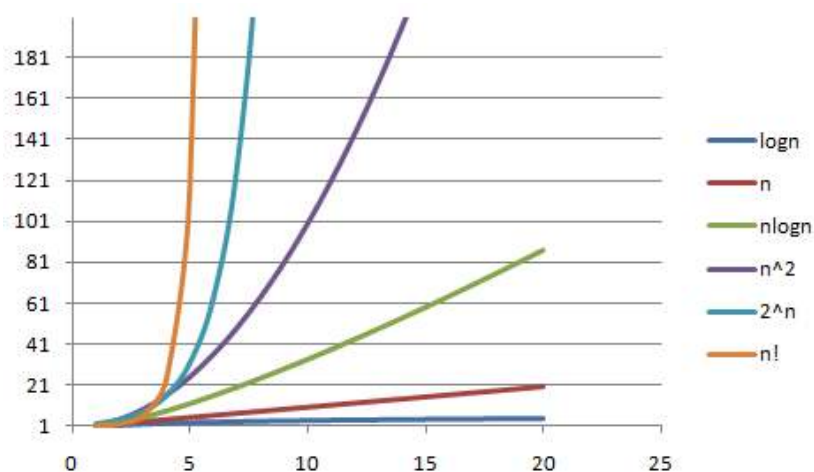
https://wenku.baidu.com/view/8a88fa54e418964bcf84b9d528ea81c758f52ed5.html?rec_flag=default&sxts=1563369616572

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2012/lecture-notes/MIT6_046JS12_lec17.pdf

<https://zhuanlan.zhihu.com/p/53085012>

- 简单问题还是困难问题
 - 日常生活 时代背景、工具、人
 - 计算机 时间、步数
- 时间复杂度
 - 并不表示确切时间
 - 大O表示法（数量级与增长趋势）
- 常见的复杂度

$f(n)$	阶	函数名
1	$O(1)$	常数函数
$2n + 1$	$O(n)$	线性函数
$2n^2 + 2n + 2$	$O(n^2)$	二次函数
$2\log 2n + 2$	$O(\log n)$	对数函数
$2n\log 2n + 2n + 2$	$O(n\log n)$	$n\log n$ 函数
$2n^3 + 2n + 2$	$O(n^3)$	三次函数
2^n	$O(2^n)$	指数函数



- P = Polynomial

多项式

arr = [...]

max = arr[0]

- N P

Non deterministic Polynemial

arr = [...]

max = 8

验证解

P时间

arr = []

median

1 排序 (nlogn)

$2arr[n/2]$

median = 8

NP

- NP complete

验证解 P

解 不是P

- 3-SAT

$x_1 x_2 \dots x_n$

True or False

CNF

$(x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \dots = \text{True}$

2^5

- $P = NP?$

[克雷数学研究所](#)高额悬赏的七个千禧年难题之一，同时也是计算机科学领域的最大难题，关系到计算机完成一项任务的速度到底有多快。

“千禧难题”之一：P (多项式算法)问题对 NP (非多项式算法)问题

“千禧难题”之二：霍奇(Hodge)猜想

“千禧难题”之三：庞加莱(Poincare)猜想

“千禧难题”之四：黎曼(Riemann)假设

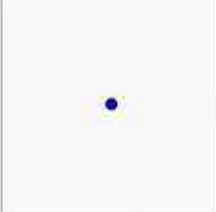
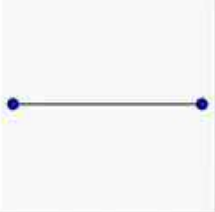
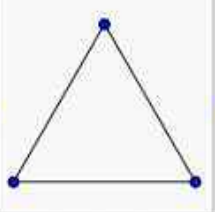
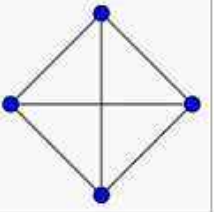
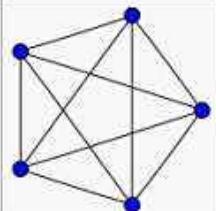
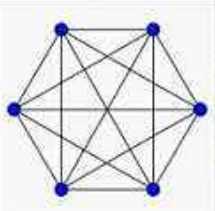
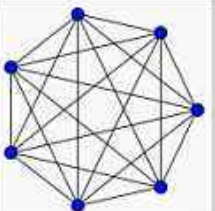
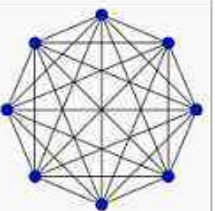
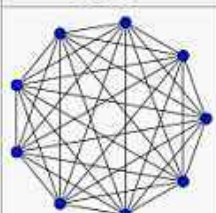

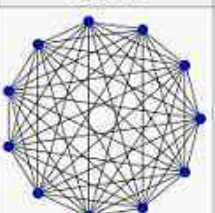
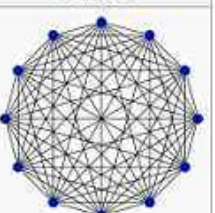
“千禧难题”之五：杨-米尔斯(Yang-Mills)存在性和质量缺口

“千禧难题”之六：纳维叶-斯托克斯(Navier-Stokes)方程的存在性与光滑性

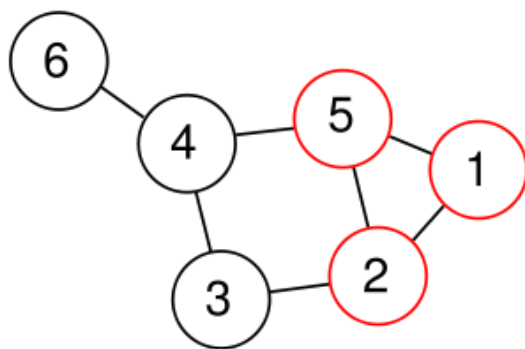
“千禧难题”之七：贝赫(Birch)和斯维纳通-戴尔(Swinnerton-Dyer)猜想

- NP C特性
 - 本质相同，问题可以互相转换（多项式时间内）
 - 一个是P，其它都是P

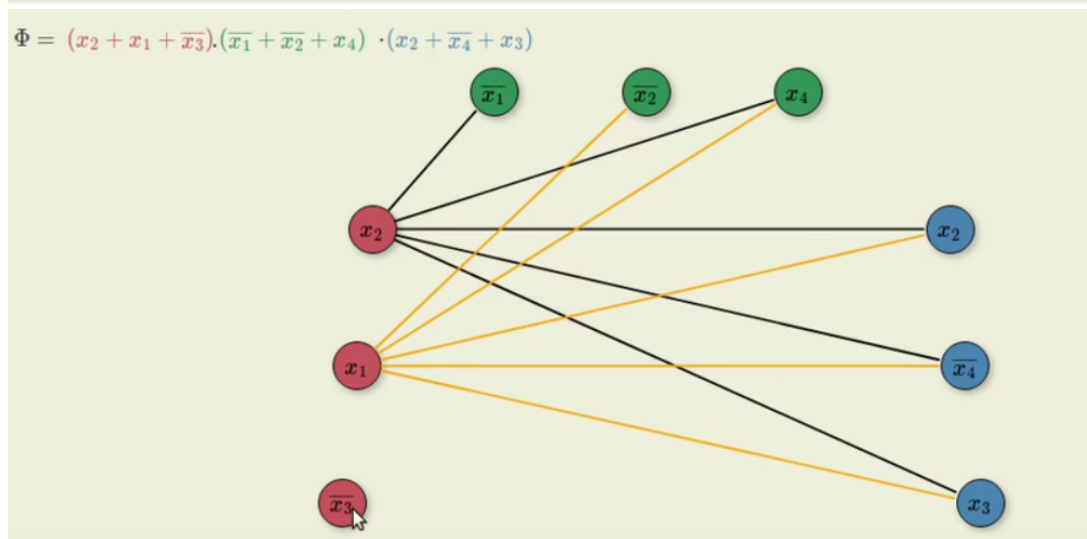
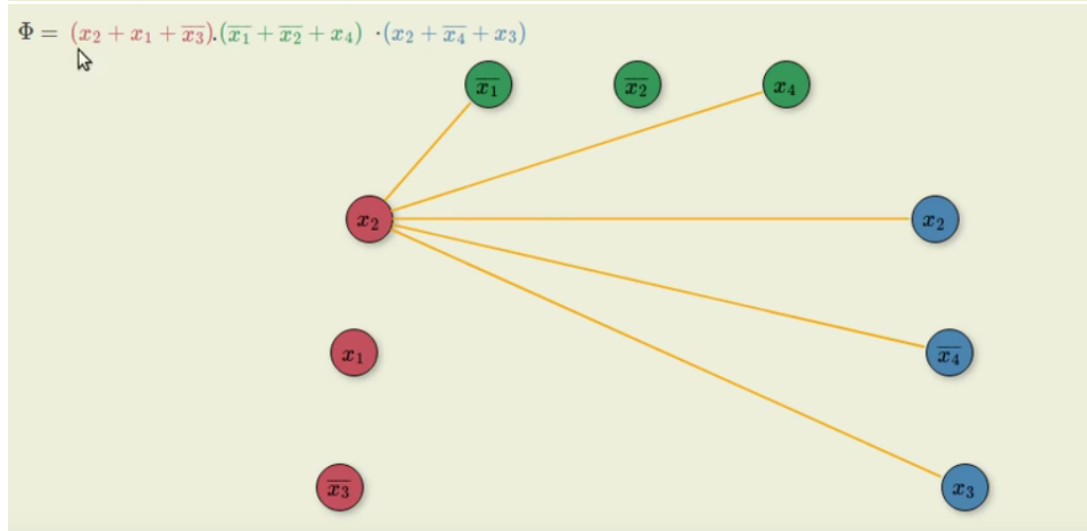
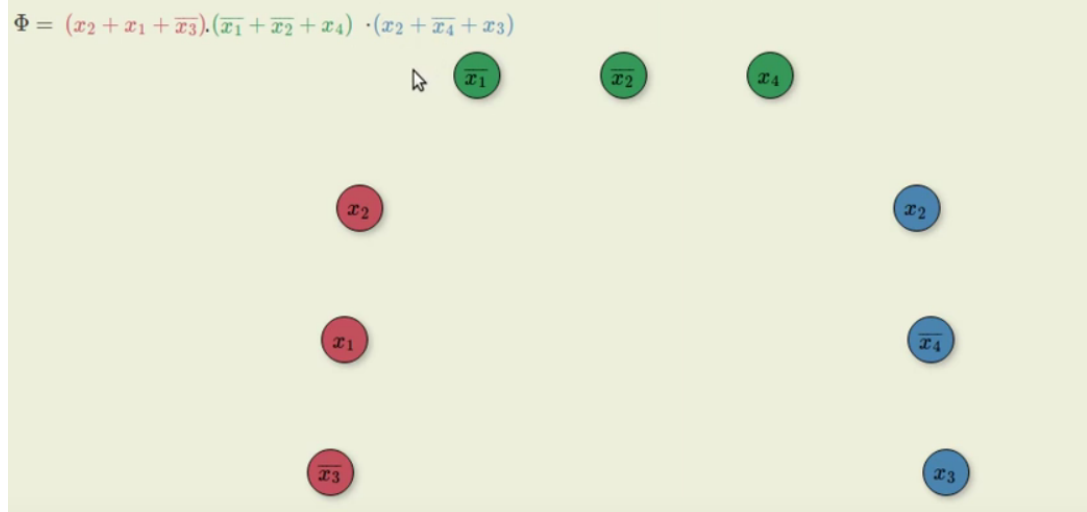
CLIQUE 团

$K_1: 0$	$K_2: 1$	$K_3: 3$	$K_4: 6$
			
$K_5: 10$	$K_6: 15$	$K_7: 21$	$K_8: 28$
			
$K_9: 36$	$K_{10}: 45$	$K_{11}: 55$	$K_{12}: 66$
			

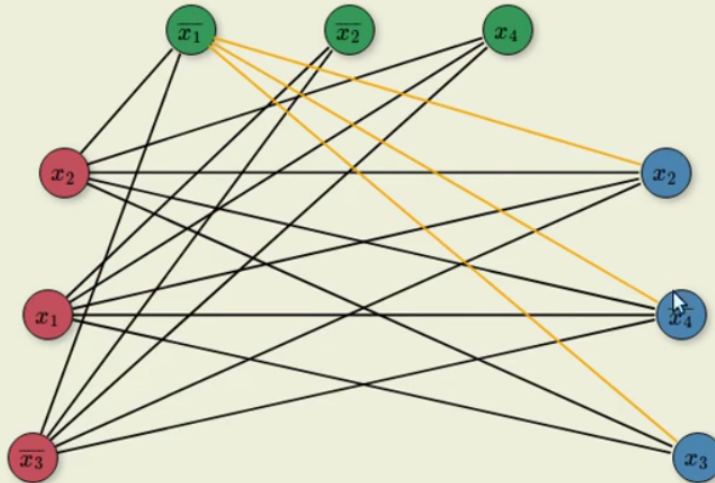
- clique problem
 - 在图中找全连接



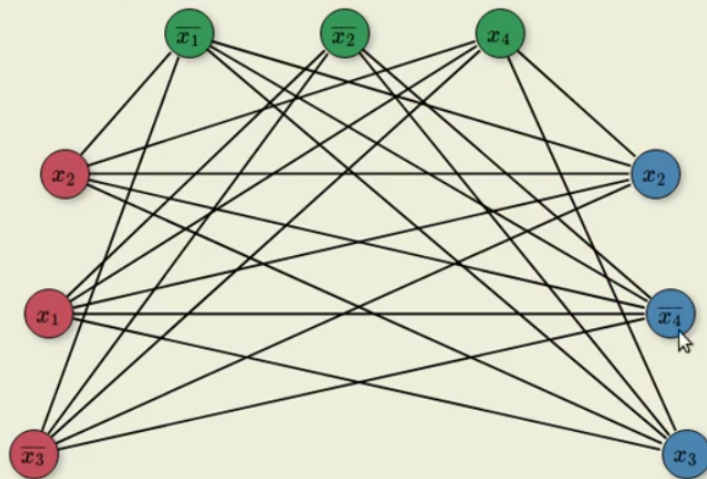
CLIQUE Problem和3-SAT (约化, Reducibility)



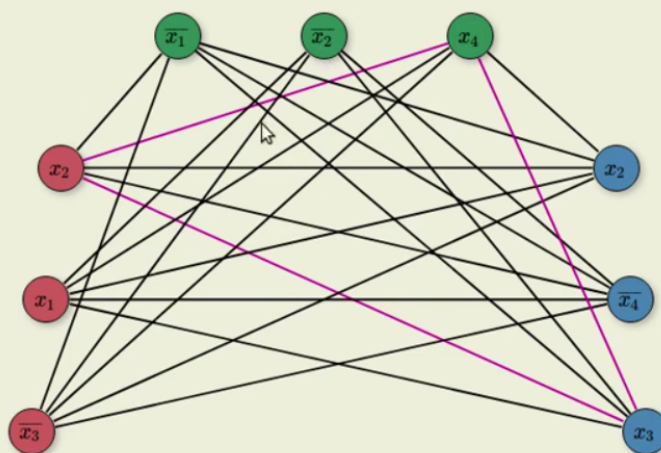
$$\Phi = (x_2 + x_1 + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_4) \cdot (x_2 + \overline{x_4} + x_3)$$



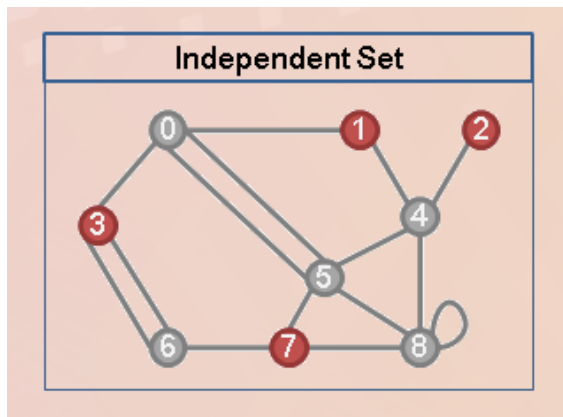
$$\Phi = (x_2 + x_1 + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_4) \cdot (x_2 + \overline{x_4} + x_3)$$



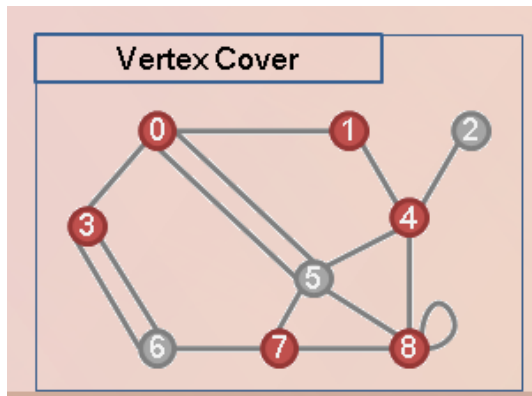
The corresponding assignment: $x_2 = \text{True}, x_3 = \text{True}, x_4 = \text{True}$



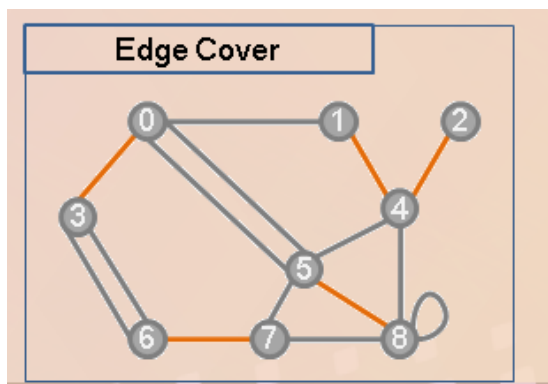
- 经典 N P C 问题
 - Independent Set



- Vertex Cover



- Edge Cover



- NP C 处理策略
 - 对问题施加限制
 - 改进指数时间算法 ($2^n \rightarrow 1.5^n$)
 - 启发式方法
 - 回溯法
 - 局部搜索 (局部搜索算法从一个初始解开始, 通过邻域动作, 产生其邻居解, 判断邻居解的质量, 根据某种策略, 来选择邻居解, 重复上述过程, 至到达终止条件。容易陷入局部最优)
 - 随机游动
 - 模拟退火
 - 遗传算法

- 平均情形的复杂性（某些算法的平均复杂度不高）

- 小故事

为了找出地球上最高的山，一群有志气的兔子们开始想办法。

- 兔子朝着比现在高的地方跳去。他们找到了不远处的最高山峰。但是这座山不一定是珠穆朗玛峰。这就是**Iterative Improvement**，它不能保证局部最优值就是全局最优值。
- 兔子喝醉了。他随机地跳了很长时间。这期间，它可能走向高处，也可能踏入平地。但是，他渐渐清醒了并朝最高方向跳去。这就是**模拟退火**。
- 兔子们知道一个兔的力量是渺小的。他们互相转告着，哪里的山已经找过，并且找过的每一座山他们都留下一只兔子做记号。他们制定了下一步去哪里寻找的策略。这就是**禁忌搜索**。
- 兔子们吃了失忆药片，并被发射到太空，然后随机落到了地球上的某些地方。他们不知道自己的使命是什么。但是，如果你过几年就杀死一部分海拔低的兔子，多产的兔子们自己就会找到珠穆朗玛峰。这就是**遗传算法**。

- 总结

