NP 完全理论

北京航空航天大学
计算机学院

提纲

- 背景介绍
- 判定问题
- P问题与多项式时间算法
- NP问题与多项式时间验证算法
- NP完全问题与多项式时间归约
- NP完全问题证明

背景介绍

- 目前掌握的算法设计技术
  - 分而治之
  - 动态规划
  - 贪心算法
- 有些问题还未找到高效的算法，如
  - 0-1背包问题
  - 旅行商问题
  - 集合覆盖问题
  - 顶点覆盖问题
  - … ..
- 需要证明不存在有效的算法

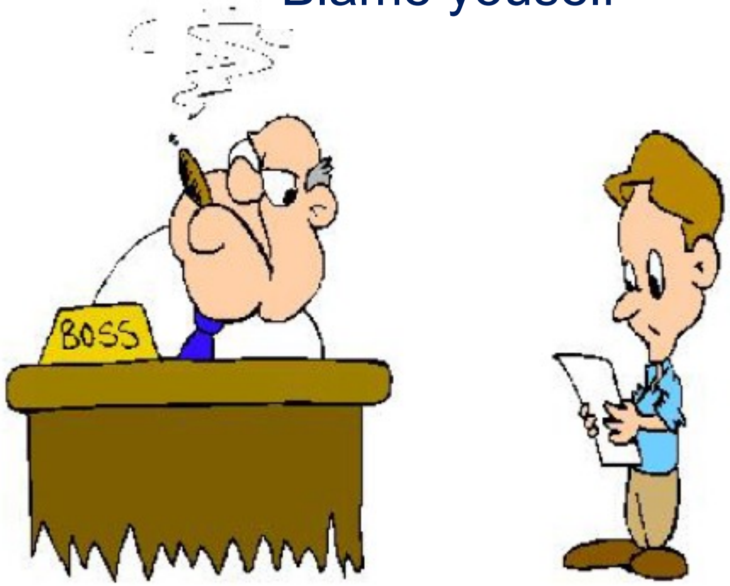
背景

- 如何证明一个问题有高效算法？
  - 设计一个高效算法

背景

- 如何证明一个问题有高效算法？
  - 设计一个高效算法
- 如何证明一个问题没有高效算法？
  - 难以证明

Blame youself



I couldn’t find a polynomial-time algorithm;
I guess I‘m too dumb.

Show that no-efficient algorithm exists

I couldn’t find a polynomial-time algorithm,
because no such algorithm exists!

背景

- 如何证明一个问题有高效算法？
  - 设计一个高效算法
- 如何证明一个问题没有高效算法？
  - 难以证明

What do you actually do？

I couldn’t find a polynomial-time algorithm,
but neither could these other smart people!

背景：NP完全理论

- NP完全问题
  - 是一类问题的集合
  - 被认为是一类难解（Intractable）问题
- 只要一个NP完全问题有高效算法，则所有NP完全问题都有高效算法
- NP问题是否有高效算法仍是一个未解决的问题（Open problem）

What do you actually do:

I couldn't find a polynomial-time algorithm, but neither could all these other smart people!

背景：NP完全理论

- NP完全问题
  - 是一类问题的集合
  - 被认为是一类难解（Intractable）问题
- 只要一个NP完全问题有高效算法，则所有NP完全问题都有高效算法
- NP问题是否有高效算法仍是一个未解决的问题（Open problem）
- 如何证明一个问题没有高效算法？
  - 证明该问题是NP难问题（NP-hard）

What do you actually do:

I couldn't find a polynomial-time algorithm,
but neither could all these other smart people!

提纲

背景介绍

判定问题

P问题与多项式时间算法

NP问题与多项式时间验证算法

NP完全问题与多项式时间归约

NP完全问题证明

判定问题（Decision problems）

- 判定问题：仅有两种答案：“是”或“否”（yes or no, 1 或 0）

判定问题（Decision problems）

- 判定问题： 仅有两种答案： “是” 或 “否” （yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

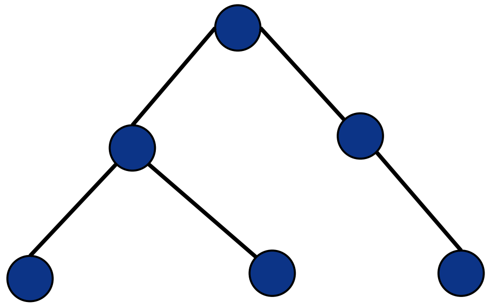
- 输入：连通无向图 G
- 问题：G 是否是一棵树？

判定问题（Decision problems）

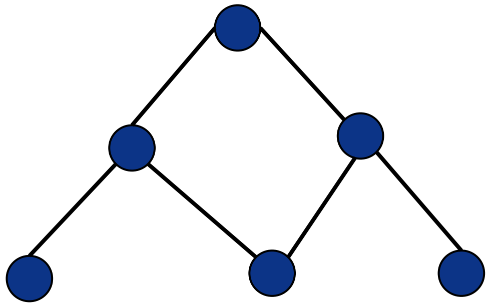
- 判定问题：仅有两种答案："是"或"否"（yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？



实例 G_1



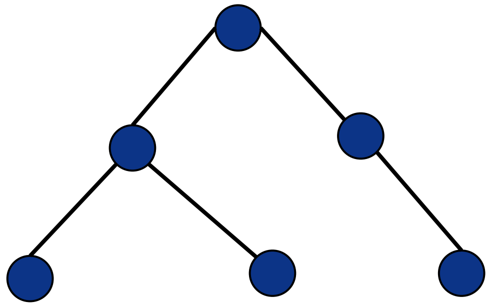
实例 G_2

判定问题（Decision problems）

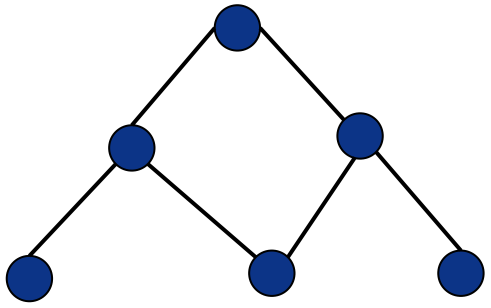
- 判定问题：仅有两种答案：“是”或“否”（yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？



实例 G_1是一棵树



实例 G_2不是一棵树

判定问题（Decision problems）

- 判定问题： 仅有两种答案： “是” 或 “否” （yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？

实例分为两类： 是树的所有连通无向图、不是树的所有连通无向图

判定问题（Decision problems）

- 判定问题： 仅有两种答案： “是” 或 “否” （yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？

实例分为两类： 是树的所有连通无向图、不是树的所有连通无向图

- 一个判定问题 Q 对应一个从 Q 的实例集 I 到 $\{0,1\}$的一个函数 f_Q

$$f_Q(x)=\begin{cases}1, & \text{若实例 } x \text{ 的答案为“是”}\\ 0, & \text{若实例 } x \text{ 的答案为“否”}\end{cases}$$

判定问题（Decision problems）

- 判定问题：仅有两种答案："是"或"否"（yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？

实例分为两类：是树的所有连通无向图、不是树的所有连通无向图

- 一个判定问题 Q 对应一个从 Q 的实例集 I 到 $\{0,1\}$的一个函数 f_Q

$$f_Q(x)=\begin{cases}1, & \text{若实例 } x \text{ 的答案为“是”}\\ 0, & \text{若实例 } x \text{ 的答案为“否”}\end{cases}$$

- 一个判定问题 Q 的实例集 $I=Y_I\cup N_I$
  - Y_I：答案为"是"的所有实例的集合
  - N_I：答案为"否"的所有实例的集合

判定问题（Decision problems）

- 判定问题：仅有两种答案："是"或"否"（yes or no, 1 或 0）

例：判断一个连通无向图是否为一棵树

- 输入：连通无向图 G
- 问题：G 是否是一棵树？

实例分为两类：是树的所有连通无向图、不是树的所有连通无向图

- 一个判定问题 Q 对应一个从 Q 的实例集 I 到 $\{0,1\}$的一个函数 f_Q

$$f_Q(x)=\begin{cases}1, & \text{若实例 } x \text{ 的答案为“是”}\\ 0, & \text{若实例 } x \text{ 的答案为“否”}\end{cases}$$

- 一个判定问题 Q 的实例集 $I=Y_I\cup N_I$
  - Y_I：答案为"是"的所有实例的集合
  - N_I：答案为"否"的所有实例的集合

NP完全理论仅考虑判定问题

提纲

- 背景介绍
- 判定问题
- P问题与多项式时间算法
- NP问题与多项式时间验证算法
- NP完全问题与多项式时间归约
- NP完全问题证明

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

抽象问题实例 $\xrightarrow{\text{编码方案}}$ 具体问题实例 $\xrightarrow{\text{算法}}$ 答案 0 或 1

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

抽象问题实例 $\xrightarrow{\text{编码方案}}$ 具体问题实例 $\xrightarrow{\text{算法}}$ 答案 0 或 1

- （抽象）问题实例
  - 生成树问题的实例包括一个加权无向图 G 和正整数 k，记为 (G, k)
  - 0-1背包问题的实例：$(1, 2, \ldots, n, w_1, \ldots, w_n, v_1, \ldots, v_n, W)$

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

抽象问题实例 $\xrightarrow{\text{编码方案}}$ 具体问题实例 $\xrightarrow{\text{算法}}$ 答案 0 或 1

- （抽象）问题实例
  - 生成树问题的实例包括一个加权无向图 G 和正整数 k，记为 (G, k)
  - 0-1背包问题的实例：$(1, 2, \ldots, n, w_1, \ldots, w_n, v_1, \ldots, v_n, W)$
- 编码方案
  - 设 Σ 是非空字母集合，编码方案 $e: I \longrightarrow \Sigma^*$ 是一个函数，将一个实例 $x \in I$ 映射到一个Σ上的字符串 $e(x)$

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

抽象问题实例	$\xrightarrow{\text{编码方案}}$	具体问题实例	$\xrightarrow{\text{算法}}$	答案 0 或 1

- （抽象）问题实例
  - 生成树问题的实例包括一个加权无向图 G 和正整数 k，记为(G, k)
  - 0-1背包问题的实例：$(1, 2, \ldots, n, w_1,\ldots, w_n, v_1, \ldots, v_n, W)$
- 二进制编码
  - 编码 $e: I \rightarrow \{0,1\}^*$ 是一个函数，将一个实例 $x \in I$ 映射到一个二进制串$e(x)$

问题实例的编码方案（Encoding scheme）

- 以计算程序可理解的方式表示问题的实例

抽象问题实例 $\xrightarrow{\text{编码方案}}$ 具体问题实例 $\xrightarrow{\text{算法}}$ 答案 0 或 1

- （抽象）问题实例
  - 生成树问题的实例包括一个加权无向图 G 和正整数 k，记为(G, k)
  - 0-1背包问题的实例：$(1, 2, \ldots, n, w_1, \ldots, w_n, v_1, \ldots, v_n, W)$
- 二进制编码
  - 编码 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将一个实例 $x \in I$ 映射到一个二进制串$e(x)$
- 具体问题：实例集为二进制串集合的问题
  - Q 是抽象问题，$e(Q)$是 Q 的编码方案 e 下的具体问题

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：合数问题

- 输入：正整数 n
- 问题：是否存在整数 $j, k > 1$，使得 $n = jk$ (即 n 是一个合数）？

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：合数问题

- 输入：正整数 n
- 问题：是否存在整数 $j, k > 1$，使得 $n = jk$ (即 n 是一个合数）？

任意正整数 n 可表示为二进制数 $a_0 a_1 a_2 \ldots a_k$，使得

$$n = a_0 2^k + a_1 2^{k-1} + a_2 2^{k-2} + \cdots + a_k 2^0,\quad k = \lceil \log_2(n+1) \rceil - 1$$

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：合数问题

- 输入：正整数 n
- 问题：是否存在整数 $j,k > 1$，使得 $n = jk$ (即 n 是一个合数）？

任意正整数 n 可表示为二进制数 $a_0a_1a_2 \ldots a_k$，使得

$$n = a_0 2^k + a_1 2^{k-1} + a_2 2^{k-2} + \cdots + a_k 2^0 ,\quad k = \lceil \log_2(n+1) \rceil - 1$$

输入规模：$\lceil \log_2(n+1) \rceil$（或$\lceil \log_2(n) \rceil$ ）

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：排序问题

- 输入：n个整数 $a_1, a_2, \dots, a_n$
- 输出：n个整数的非递减排序

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：排序问题

- 输入：n个整数 $a_1, a_2, \ldots, a_n$
- 输出：n个整数的非递减排序

使用固定长度编码：令 $m = \lceil \log_2 \max(|a_i| + 1) \rceil$，则

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例：排序问题

- 输入：n个整数 $a_1, a_2, \dots, a_n$
- 输出：n个整数的非递减排序

使用固定长度编码：令 $m = \lceil \log_2 \max(|a_i| + 1) \rceil$，则

输入规模为 nm

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例： 把图 $G = (V, E)$表示为邻接矩阵 $A = (a_{ij})_{n\times n}$ ，

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例： 把图 $G = (V, E)$表示为邻接矩阵 $A = (a_{ij})_{n \times n}$ ，

则 G 可编码为长度为 n^2 的二进制串：

$$a_{11} \dots a_{1n} a_{21} \dots a_{2n} \dots a_{n1} \dots a_{nn}$$

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。

例： 把图 $G = (V, E)$表示为邻接矩阵 $A = (a_{ij})_{n \times n}$，

则 G 可编码为长度为 n^2 的二进制串：

$$a_{11} \ldots a_{1n} a_{21} \ldots a_{2n} \ldots a_{n1} \ldots a_{nn}$$

输入规模： n^2

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。
- 问题 Q 和编码方案 e 把 $\{0,1\}^*$ 划分成三类字符串
  - 不是问题 Q 的实例的编码
  - 答案为“否”的实例的编码
  - 答案为“是”的实例的编码

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。
- 问题 Q 和编码方案 e 把 $\{0,1\}^*$ 划分成三类字符串
  - 不是问题 Q 的实例的编码
  - 答案为“否”的实例的编码
  - 答案为“是”的实例的编码

二进制编码方案

- 编码方案 $e: I \longrightarrow \{0,1\}^*$ 是一个函数，将判定问题 Q 的一个实例 $x \in I$ 映射到一个二进制串$e(x)$，称为 x 的编码
- 问题实例规模
  - 一个实例 x 的规模为 x 的二进制编码$e(x)$的长度，记为$|x|$。
- 问题 Q 和编码方案 e 把 $\{0,1\}^*$ 划分成三类字符串
  - 不是问题 Q 的实例的编码
  - 答案为“否”的实例的编码（ $Q(x) = 0$ ）
  - 答案为“是”的实例的编码（ $Q(x) = 1$ ）

问题 Q 可描述为语言 $L = \{x \in \{0,1\}^* | Q(x) = 1\}$，即答案为“是”的实例的编码的集合。

多项式时间可解：具体问题

- 给定一个具体问题的实例 $x \in I$，且 $|x| = n$。

若算法 A可在 $O(T(n))$ 时间内得到该具体问题的解，则称算法 A 在时间$O(T(n))$内解决了该具体问题。

多项式时间可解：具体问题

- 给定一个具体问题的实例 $x \in I$，且 $|x| = n$。

若算法 A可在 $O(T(n))$ 时间内得到该具体问题的解，则称算法 A 在时间$O(T(n))$内解决了该具体问题。

- 如果对某个常数 k，存在一个算法能在时间 $O(n^k)$内求解出某具体问题，则称该具体问题是多项式时间可解的

多项式时间可解：具体问题

- 给定一个具体问题的实例 $x \in I$，且 $|x| = n$。

若算法 A可在 $O(T(n))$ 时间内得到该具体问题的解，则称算法 A 在时间$O(T(n))$内解决了该具体问题。

- 如果对某个常数 k，存在一个算法能在时间 $O(n^k)$内求解出某具体问题，则称该具体问题是多项式时间可解的

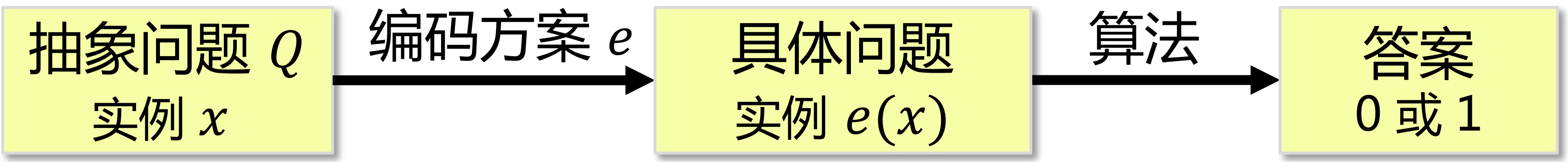
多项式时间可解：抽象问题

- 设 Q 是一个抽象问题，其实例集为 I， 则一个二进制编码方案 $e\colon I \longrightarrow \{0,1\}^*$ 可以导出与 Q 相关的具体问题，记为 $e(Q)$，其实例集为$e(I) = \{e(x)|x \in I\}$

多项式时间可解：抽象问题

- 设 Q 是一个抽象问题，其实例集为 I， 则

  一个二进制编码方案 $e: I \longrightarrow \{0,1\}^*$ 可以导出与 Q 相关的具体问题，记为 $e(Q)$，其实例集为$e(I) = \{e(x)|x \in I\}$



多项式时间可解：抽象问题

- 设 Q 是一个抽象问题，其实例集为 I， 则

  一个二进制编码方案 $e: I \longrightarrow \{0,1\}^*$ 可以导出与 Q 相关的具体问题，记为 $e(Q)$，其实例集为$e(I) = \{e(x)|x \in I\}$

$$\boxed{\begin{array}{c}\text{抽象问题 } Q \\ \text{实例 } x\end{array}} \xrightarrow{\text{编码方案 } e} \boxed{\begin{array}{c}\text{具体问题} \\ \text{实例 } e(x)\end{array}} \xrightarrow{\text{算法}} \boxed{\begin{array}{c}\text{答案} \\ 0 \text{ 或 } 1\end{array}}$$

- 注意：实例规模 $|x| = n$ 依赖于编码方式

- 对同一个问题的两个编码方案 e_1, e_2， 是否有

$$e_1(Q) \in \mathrm{P} \Leftrightarrow e_1(Q) \in \mathrm{P}\ ?$$

举例说明

- 0-1背包问题的动态规划算法
  - 算法复杂度：$O(nW)$，
  - n 为物品个数，W为背包容量
- 问题：是否为多项式时间算法？

举例说明

- 0-1背包问题的动态规划算法
  - 算法复杂度：$O(nW)$，
  - n 为物品个数，W为背包容量
- 问题：是否为多项式时间算法？
  - 一进制编码：
  - 二进制编码：

举例说明

- 0-1背包问题的动态规划算法
  - 算法复杂度：$O(nW)$，
  - n 为物品个数，W为背包容量
- 问题：是否为多项式时间算法？
  - 一进制编码：W 编码为由 W个 1 组成的串
  - 二进制编码：W 编码长度为 $k = \lceil \log_2(n+1) \rceil$

举例说明

- 0-1背包问题的动态规划算法
  - 算法复杂度：$O(nW)$，
  - n 为物品个数，W为背包容量
- 问题：是否为多项式时间算法？
  - 一进制编码：W 编码为由 W个 1 组成的串

    算法复杂度为 $O(nW)$
  - 二进制编码：W 编码长度为 $k = \lceil \log_2(W+1) \rceil$

    算法复杂度为 $O(n2^k)$

举例说明

- 0-1背包问题的动态规划算法（伪多项式时间算法）
  - 算法复杂度：$O(nW)$，
  - n 为物品个数，W为背包容量
- 问题：是否为多项式时间算法？
  - 一进制编码：W 编码为由 W个 1 组成的串

    算法复杂度为 $O(nW)$
  - 二进制编码：W 编码长度为 $k = \lceil \log_2(W+1) \rceil$

    算法复杂度为 $O(n2^k)$

与编码无关的复杂度

- 问题的性质应与编码方案无关
  - 如果 e_1 和 e_2 是问题 Q 的两个合理的编码方案，那么性质对$e_1(Q)$与$e_2(Q)$同时成立
    - $e_1(Q) \in \mathrm{P}$ 当且仅当 $e_2(Q) \in \mathrm{P}$

与编码无关的复杂度

- 问题的性质应与编码方案无关
  - 如果 e_1 和 e_2 是问题 Q 的两个合理的编码方案，那么性质对$e_1(Q)$与$e_2(Q)$同时成立
    - $e_1(Q) \in \mathrm{P}$ 当且仅当 $e_2(Q) \in \mathrm{P}$

需要使用“合理”的编码方案，使复杂度与编码方案无关

与编码无关的复杂度

- 问题的性质应与编码方案无关
  - 如果 e_1 和 e_2 是问题 Q 的两个合理的编码方案，那么性质对$e_1(Q)$与$e_2(Q)$同时成立
    - $e_1(Q) \in \mathrm{P}$ 当且仅当 $e_2(Q) \in \mathrm{P}$

需要使用“合理”的编码方案，使复杂度与编码方案无关

- 合理编码方案
  - 合理是指在编码中不故意使用许多冗余的字符
  - 当采用合理的编码时，输入的规模都是“多项式相关”的

多项式相关的编码方案

对某个问题 Q 的实例集 I，

- 多项式相关的编码方案
  - 如果存在两个多项式时间可计算的函数 f 和f'，满足对任意的 $x \in I$，有

$$f\big(e_1(x)\big) = e_2(x)\text{，且 } f'\big(e_2(x)\big) = e_1(x)\text{，}$$

  则称两种编码 e_1 和 e_2 是多项式相关的

- 多项式相关的编码（输入）长度
  - 设e_1 和 e_2 是 Q 的两个多项式相关的编码方案，则存在两个多项式 p 和 p'，满足对任意的 $x \in I$，有

$$|e_2(x)| \le p(|e_1(x)|)\text{，且} |e_1(x)| \le p'(|e_2(x)|)\text{，}$$

合理编码

- 合理是指在编码中不故意使用许多冗余的字符
- 当采用合理的编码时，输入的规模都是多项式相关的

例：对简单无向图 $G=(V,E)$进行编码。

合理编码

- 合理是指在编码中不故意使用许多冗余的字符
- 当采用合理的编码时，输入的规模都是多项式相关的

例：对简单无向图 $G = (V, E)$进行编码。

- G用邻接矩阵表示，进行二进制编码：

$$|G| = |V|^2$$

合理编码

- 合理是指在编码中不故意使用许多冗余的字符
- 当采用合理的编码时，输入的规模都是多项式相关的

例：对简单无向图 $G=(V,E)$进行编码。

- G用邻接矩阵表示，进行二进制编码：

$$|G| = |V|^2$$

- 对 G 中每个顶点和边进行固定长度编码：

$$|G| = (|V|+|E|)log_2(|V|+|E|+1)$$

合理编码

- 合理是指在编码中不故意使用许多冗余的字符
- 当采用合理的编码时，输入的规模都是多项式相关的

例：对简单无向图 $G = (V, E)$进行编码。

- G用邻接矩阵表示，进行二进制编码：

$$|G| = |V|^2$$

- 对 G 中每个顶点和边进行固定长度编码：

$$|G| = (|V| + |E|)log_2(|V| + |E| + 1)$$

可证：$|V|^2$，$(|V| + |E|)log_2(|V| + |E| + 1)$，$|V| + |E|$是两两多项式相关的。

因此，通常使用$|V| + |E|$ 作为输入规模。

合理编码

- 合理是指在编码中不故意使用许多冗余的字符
- 当采用合理的编码时，输入的规模都是多项式相关的

例：对简单无向图 $G=(V,E)$进行编码。

- G用邻接矩阵表示，进行二进制编码：

$$|G|=|V|^2$$

- 对 G 中每个顶点和边进行固定长度编码：

$$|G|=(|V|+|E|)log_2(|V|+|E|+1)$$

可证：$|V|^2$，$(|V|+|E|)log_2(|V|+|E|+1)$，$|V|+|E|$是两两多项式相关的。

因此，通常使用$|V|+|E|$ 作为输入规模。

- 一进制编码与二制编码不是多项式相关的

P类问题

- 定理：设 Q 是一个抽象问题，I 是 Q 的实例集， e_1 和 e_2 是 Q 的两个多项式相关的编码方案，则

$$e_1(Q) \in \mathrm{P} \text{ 当且仅当 } e_2(Q) \in \mathrm{P}$$

P类问题

- 定理：设 Q 是一个抽象问题，I 是 Q 的实例集， e_1 和 e_2 是 Q 的两个多项式相关的编码方案，则

$$e_1(Q) \in \mathrm{P} \text{ 当且仅当 } e_2(Q) \in \mathrm{P}$$

- 定义：具有多项式时间算法的判定问题称为 P问题。
- P类：所有P问题的集合

P类问题举例

- 生成树判定问题（DMST）
- 判断一个无向图（有向图）是否有循环
- 部分背包问题的判定问题
- … …

提纲

- 背景介绍
- 判定问题
- P问题与多项式时间算法
- NP问题与多项式时间验证算法
- NP完全问题与多项式时间归约
- NP完全问题证明

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：生成树判定问题DST：

- 输入：加权无向图 G，正整数 k
- 问题：G 是否存在边权和不超过 k 的生成树 ？

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：生成树判定问题DST：

- 输入：加权无向图 G，正整数 k
- 问题：G 是否存在边权和不超过 k 的生成树 ？

如果对于输入(G,k)，答案为“是” ，则必存在G 的边集 $E' \subseteq E$，满足

- E' 是 图 G 的一棵生成树；
- E' 的边权和不超过 k.

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：生成树判定问题DST：

- 输入：加权无向图 G，正整数 k
- 问题：G 是否存在边权和不超过 k 的生成树 ？

如果对于输入(G,k)，答案为 “是” ，则必存在G 的边集 $E' \subseteq E$，满足

- E' 是 图 G 的一棵生成树；
- E' 的边权和不超过 k.

E' 可看作是证明 G存在边权和不超过 k的生成树的一个证书(certificate)

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：0-1背包判定问题

- 输入： n 个物品的集合$X = \{1,2,\dots,n\}$，其中每个物品 i 具有重w_i和价值v_i，$i = 1,2,\dots,n$，和背包容量 W，正整数 V
- 问题：是否存在能装入背包的价值和至少为 V 的物品子集？

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：0-1背包判定问题

- 输入： n 个物品的集合$X=\{1,2,\dots,n\}$，其中每个物品 i 具有重w_i和价值v_i, $i=1,2,\dots,n$，和背包容量 W，正整数 V
- 问题：是否存在能装入背包的价值和至少为 V 的物品子集？

如果对于输入(G,k)，答案为“是”，则必存在物品子集$X'\subseteq X$，满足

- $\sum_{i\in X'} w_i \le \mathrm{W}$
- $\sum_{i\in X'} v_i \ge V$

判定问题与证书

- 一个判定问题的答案往往依赖于判断具有某个条件的事物是否存在

例：0-1背包判定问题

- 输入： n 个物品的集合$X = \{1,2,\ldots,n\}$，其中每个物品 i 具有重w_i和价值v_i，$i = 1,2,\ldots,n$，和背包容量 W，正整数 V
- 问题：是否存在能装入背包的价值和至少为 V 的物品子集？

如果对于输入(G,k)，答案为 “是” ，则必存在物品子集$X' \subseteq X$ ，满足

- $\sum_{i\in X'} w_i \le W$
- $\sum_{i\in X'} v_i \ge V$

X' 可看作是证明存在能装入背包的价值和至少为 V的物品子集的一个证书。

判定问题与证书

- 例：哈密顿回路问题
  - 输入：无向图 $G = (V, E)$
  - 问题：图 G 是否为哈密顿图，即是否具有一条哈密顿回路？

其中，无向图 G 中的一个哈密顿回路是通过V中每个顶点一次且仅一次的回路；具有哈密顿回路的图称为哈密顿图

判定问题与证书

- 例：哈密顿回路问题
  - 输入：无向图 $G = (V, E)$
  - 问题：图 G 是否为哈密顿图，即是否具有一条哈密顿回路？

其中，无向图 G 中的一个哈密顿回路是通过V中每个顶点一次且仅一次的回路；具有哈密顿回路的图称为哈密顿图

如果对于输入G，答案为“是”，则必存在顶点序列ρ，满足：

- 除了第一个顶点在末尾重复出现一次外，序列 ρ 恰好包含 V 中每个顶点一次；
- ρ 形成 一个回路，即每一对连续顶点及首、尾顶点之间都存在一条边。

判定问题与证书

- 例：哈密顿回路问题
  - 输入：无向图 $G = (V, E)$
  - 问题：图 G 是否为哈密顿图，即是否具有一条哈密顿回路？

其中，无向图 G 中的一个哈密顿回路是通过V中每个顶点一次且仅一次的回路；具有哈密顿回路的图称为哈密顿图

如果对于输入G，答案为“是”，则必存在顶点序列ρ，满足：

- 除了第一个顶点在末尾重复出现一次外，序列 ρ 恰好包含 V 中每个顶点一次；
- ρ 形成 一个回路，即每一对连续顶点及首、尾顶点之间都存在一条边。

ρ 可看作是证明图G 是哈密顿图的证书。

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x, y) = 1$ ，则算法 A 验证了输入串 x

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x,y)=1$，则算法 A 验证了输入串 x
- NP问题

  一个判定问题 Q 是NP问题，记作 $Q\in \mathrm{NP}$，

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x,y)=1$，则算法 A 验证了输入串 x
- NP问题

一个判定问题 Q 是NP问题，记作 $Q\in \text{NP}$，当且仅当 存在一个两输入的多项式时间验证算法 A 和常数 c 使得：（设 Q 的实例集为 I）

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x, y) = 1$，则算法 A 验证了输入串 x
- NP问题

　　一个判定问题 Q 是NP问题，记作 $Q \in \text{NP}$，当且仅当 存在一个两输入的多项式时间验证算法 A 和常数 c 使得：（设 Q 的实例集为 I）

对任意实例 $x \in Y_I$，存在一个证书 y且 $|y| = O(|x|^c)$，满足 $A(x, y) = 1$。

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x,y)=1$，则算法 A 验证了输入串 x
- NP问题

一个判定问题 Q 是NP问题，记作 $Q \in \mathrm{NP}$，当且仅当 存在一个两输入的多项式时间验证算法 A 和常数 c 使得：（设 Q 的实例集为 I）

对任意实例 $x \in Y_I$，存在一个证书 y且 $|y| = O(|x|^c)$，满足 $A(x,y)=1$。

称 算法 A 在多项式时间内验证了判定问题 Q。

多项式可验证与NP问题

- 验证算法 A
  - 以 输入串 x 和 称为 “证书” 的二进制串 y为输入
  - 如果存在一个证书 y 满足 $A(x, y) = 1$，则算法 A 验证了输入串 x
- NP问题

一个判定问题 Q 是NP问题，记作 $Q \in \mathrm{NP}$，当且仅当 存在一个两输入的多项式时间验证算法 A 和常数 c 使得：（设 Q 的实例集为 I）

对任意实例 $x \in Y_I$，存在一个证书 y且 $|y| = O(|x|^c)$，满足 $A(x, y) = 1$。

称 算法 A 在多项式时间内验证了判定问题 Q。

NP问题举例

- 生成树判定问题 （DMST）
- 布尔公式的可满足性问题（SAT）
- 3-CNF的可满足性问题（3SAT）
- 0-1 背包问题（0-1 Knapsack）

NP问题举例：DMST

- DMST∈NP
- 定理：P ⊆ NP
  - 多项式可解的判定问题一定是多项式可验证的。

NP问题举例：SAT

- 布尔公式：由布尔变量和逻辑运算 ¬，∨，∧ 构成的逻辑公式，其中，逻辑运算定义如下：

x_1	x_2	$\neg x_1$	$x_1 \vee x_2$	$x_1 \wedge x_2$
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

例：$\varphi_1 = (x_1 \land (x_2 \lor \neg x_3)) \lor (\neg x_2 \land x_3 \land \neg x_1)$，其真值表为

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$(x_1 \land (x_2 \lor \neg x_3))$	$(\neg x_2 \land x_3 \land \neg x_1)$	φ_1
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	0	1

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

例：$\varphi_1 = (x_1 \wedge (x_2 \vee \neg x_3)) \vee (\neg x_2 \wedge x_3 \wedge \neg x_1)$，其真值表为

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$(x_1 \wedge (x_2 \vee \neg x_3))$	$(\neg x_2 \wedge x_3 \wedge \neg x_1)$	φ_1
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	0	1

φ_1是可满足的

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

例：$\varphi_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$ ，

其真值表为

x_1	x_2	$x_1 \vee x_2$	$\neg x_1 \vee x_2$	$x_1 \vee \neg x_2$	$\neg x_1 \vee \neg x_2$	φ_2
0	0	0	1	1	1	0
0	1	1	1	0	1	0
1	0	1	0	1	1	0
1	1	1	1	1	0	0

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

例：$\varphi_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$，

其真值表为 φ_2 不可满足

x_1	x_2	$x_1 \vee x_2$	$\neg x_1 \vee x_2$	$x_1 \vee \neg x_2$	$\neg x_1 \vee \neg x_2$	φ_2
0	0	0	1	1	1	0
0	1	1	1	0	1	0
1	0	1	0	1	1	0
1	1	1	1	1	0	0

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。
- SAT $\in$NP

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。
- SAT $\in$NP

证明：给定公式 φ 的一个真值赋值 μ，

下面证明可在多项式时间内验证 φ 在 真值赋值 μ 下是否为真。

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。
- SAT $\in$NP

证明：给定公式 φ 的一个真值赋值 μ，

下面证明可在多项式时间内验证 φ 在 真值赋值 μ 下是否为真。

假设公式φ 的长度为 n（包括所有变量、逻辑算子、括号），

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。
- SAT ∈NP

证明：给定公式 φ 的一个真值赋值 μ，

下面证明可在多项式时间内验证 φ 在 真值赋值 μ 下是否为真。

假设公式φ 的长度为 n（包括所有变量、逻辑算子、括号），

因此最多需要 n 次 真值评估（真值计算），且每一次为常数时间。

NP问题举例：SAT

- SAT
  - 输入：一个布尔公式 φ，其中 φ 中的所有布尔变量来自集合 X
  - 问题：φ 是否是可满足的，即是否存在一个真值赋值使得 φ 为真。

- SAT $\in$NP

证明：给定公式 φ 的一个真值赋值 μ，

下面证明可在多项式时间内验证 φ 在 真值赋值 μ 下是否为真。

假设公式φ 的长度为 n（包括所有变量、逻辑算子、括号），

因此最多需要 n 次 真值评估（真值计算），且每一次为常数时间。

所以，验证需要时间为 $O(n)$。

因此，SAT $\in$NP。

NP问题举例：3SAT

- 3SAT
  - 输入：一个3-CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集合 $X = \{x_1, x_2, \ldots, x_m\}$ 的合取范式，且对任意 $i \in [1, n]$, $C_i = l_1^i \vee l_2^i \vee l_3^i$，$l_j^i \in X$ 或 $\neg l_j^i \in X$
  - 问题：φ是否为可满足的，即是否存在一个真值赋值使得 φ 为真。

NP问题举例：3SAT

- 3SAT
  - 输入：一个3-CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集合 $X = \{x_1, x_2, \ldots, x_m\}$ 的合取范式，且对任意 $i \in [1, n]$, $C_i = l_1^i \vee l_2^i \vee l_3^i$，$l_j^i \in X$ 或 $\neg l_j^i \in X$
  - 问题：φ是否为可满足的，即是否存在一个真值赋值使得 φ 为真。

例： 令 $X = \{x_1, x_2, x_3\}$,

$$\varphi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

NP问题举例：3SAT

- 3SAT
  - 输入：一个3-CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集合 $X = \{x_1, x_2, \ldots, x_m\}$ 的合取范式，且对任意 $i \in [1, n]$, $C_i = l_1^i \vee l_2^i \vee l_3^i$，$l_j^i \in X$ 或 $\neg l_j^i \in X$
  - 问题：φ是否为可满足的，即是否存在一个真值赋值使得 φ 为真。

例： 令 $X = \{x_1, x_2, x_3\}$,

$$\varphi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

真值赋值 $\mu(x_1, x_2, x_3) = (1, 1, 0)$，有 $\mu(\varphi) = 1$。

因此，φ 是可满足的。

NP问题举例：3SAT

- 3SAT
  - 输入：一个3-CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集合 $X = \{x_1, x_2, \ldots, x_m\}$ 的合取范式，且对任意 $i \in [1, n]$, $C_i = l_1^i \vee l_2^i \vee l_3^i$，$l_j^i \in X$ 或 $\neg l_j^i \in X$
  - 问题：φ是否为可满足的，即是否存在一个真值赋值使得 φ 为真。
- 3SAT $\in$NP

NP问题举例：3SAT

- 3SAT
  - 输入： 一个3-CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集合 $X = \{x_1, x_2, \ldots, x_m\}$ 的合取范式，且对任意 $i \in [1, n]$, $C_i = l_1^i \vee l_2^i \vee l_3^i$，$l_j^i \in X$ 或 $\neg l_j^i \in X$
  - 问题：φ是否为可满足的，即是否存在一个真值赋值使得 φ 为真。
- 3SAT ∈NP
  - 3SAT 是 SAT的特例
  - SAT ∈NP

NP问题举例：0-1 Knapsack

- 0-1 Knapsack
  - 输入： n 个物品的集合$X = \{1,2, \dots, n\}$，其中每个物品 i 具有重w_i和价值v_i, $i = 1, 2, \dots, n$，和背包容量 W，正整数 V
  - 问题：是否存在能装入背包的价值和至少为 V 的物品子集？
- 0-1 Knapsack ∈NP

NP问题举例：0-1 Knapsack

- 0-1 Knapsack
  - 输入：n 个物品的集合$X = \{1,2, \dots, n\}$，其中每个物品 i 具有重w_i和价值v_i, $i = 1, 2, \dots, n$，和背包容量 W，正整数 V
  - 问题：是否存在能装入背包的价值和至少为 V 的物品子集？

- 0-1 Knapsack ∈NP

证明：给定一个物品子集 $X' \subseteq X$，

NP问题举例：0-1 Knapsack

- 0-1 Knapsack
  - 输入： n 个物品的集合$X = \{1,2,\dots,n\}$，其中每个物品 i 具有重w_i和价值v_i, $i = 1,2,\dots,n$，和背包容量 W，正整数 V
  - 问题：是否存在能装入背包的价值和至少为 V 的物品子集？

- 0-1 Knapsack ∈NP

证明：给定一个物品子集 $X' \subseteq X$，显然，可在多项式时间内验证以下两个条件是否成立：

(1) $\sum_{i\in X'} w_i \le W$，且

(2) $\sum_{i\in X'} v_i \ge V$

因此，0-1背包问题是NP问题。

提纲

背景介绍

判定问题与优化问题

P问题与多项式时间算法

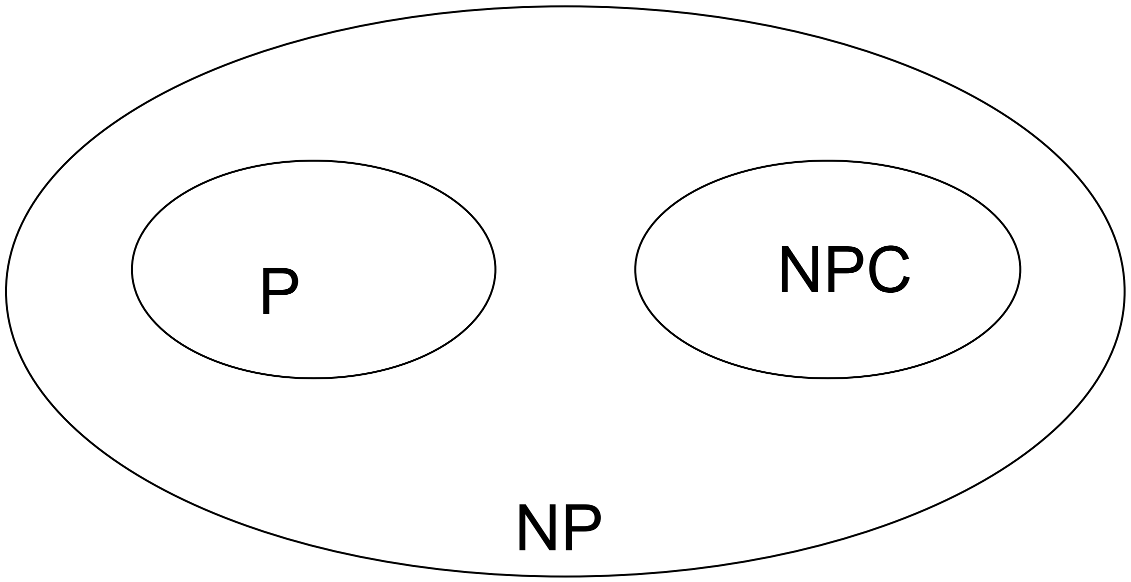
NP问题与多项式时间验证算法

NP完全问题与多项式时间归约

NP完全问题证明

NP-完全理论

- 多项式时间归约
- NP完全类（NPC）
- NP完全问题



多项式时间归约

- 设 Q_1与Q_2 是两个判定问题，且实例集分别为 I_1 与 I_2。

  如果存在一个多项式可计算的变换 f，使得

对于 Q_1 的任意实例 x，

  - $f(x)$ 是 Q_2 的一个实例，且
  - $x \in Y_{I_1}$ 当且仅当 $f(x) \in Y_{I_2}$，

则称 Q_1 多项式归约至 Q_2，记为$Q_1 \leq_P Q_2$，并称 f 为从 Q_1 到 Q_2 的多项式时间归约。

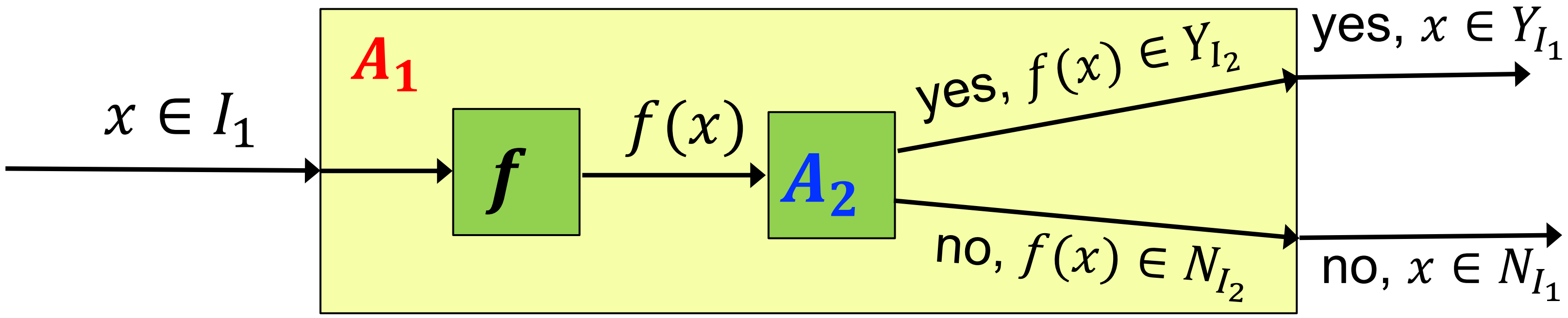
多项式时间归约的应用：$Q_1 \leq_P Q_2$

多项式时间归约的应用：$Q_1 \leq_P Q_2$

- 假设 A_2 为解决问题 Q_2 的算法，则可构建解决 Q_1 的算法 A_1

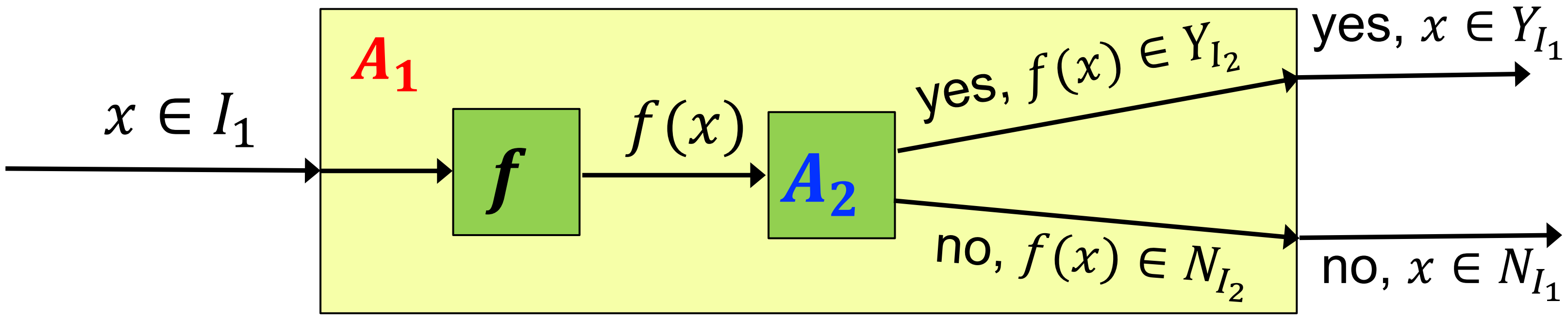
多项式时间归约的应用：$Q_1 \leq_P Q_2$

- 假设 A_2 为解决问题 Q_2 的算法，则可构建解决 Q_1 的算法 A_1



多项式时间归约的应用：$Q_1 \leq_P Q_2$

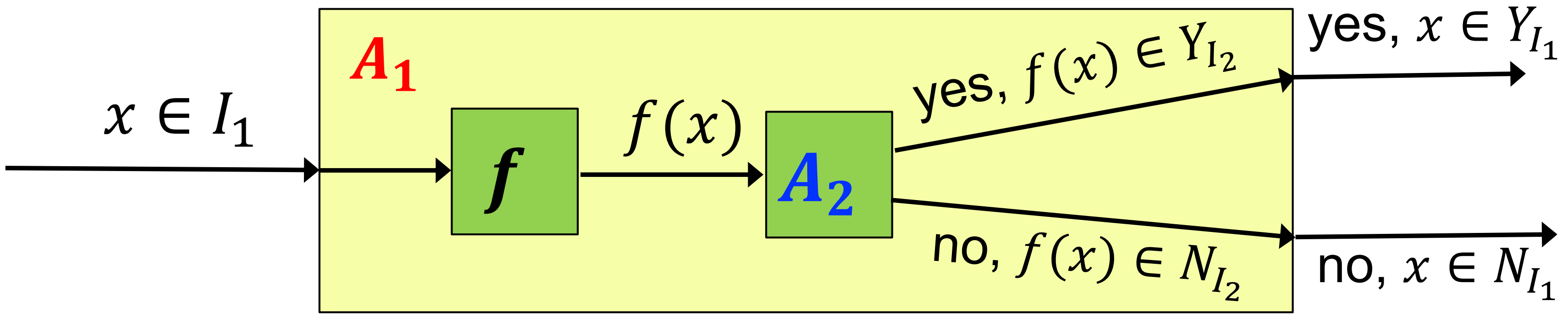
- 假设 A_2 为解决问题 Q_2 的算法，则可构建解决 Q_1 的算法 A_1



- 若 $Q_1 \leq_P Q_2$，则 问题 Q_1 不会比 Q_2 难

多项式时间归约的应用：$Q_1 \le_P Q_2$

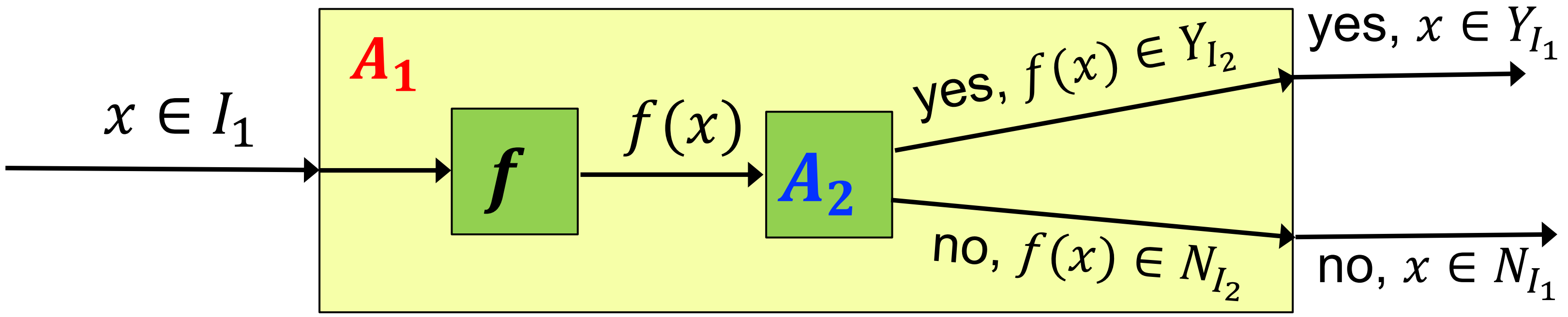
- 假设 A_2 为解决问题 Q_2 的算法，则可构建解决 Q_1 的算法 A_1



- 若 $Q_1 \le_P Q_2$，则 问题 Q_1 不会比 Q_2 难
- 若 A_2 是多项式时间算法，则A_1必为多项式时间算法

多项式时间归约的应用：$Q_1 \leq_P Q_2$

- 假设 A_2 为解决问题 Q_2 的算法，则可构建解决 Q_1 的算法 A_1



- 若 $Q_1 \leq_P Q_2$，则 问题 Q_1 不会比 Q_2 难
- 若 A_2 是多项式时间算法，则A_1必为多项式时间算法
- 定理：若 $Q_1 \leq_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

多项式时间归约

- 定理：若 $Q_1 \leq_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \leq_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

多项式时间归约

- 定理：若 $Q_1 \le_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \le_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

由于$Q_2 \in \mathrm{P}$ 知， Q_2有多项式时间算法，设为 A_2。

多项式时间归约

- 定理：若 $Q_1 \leq_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \leq_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

由于$Q_2 \in \mathrm{P}$ 知， Q_2有多项式时间算法，设为 A_2。

可如下构造 Q_1的多项式时间算法 A_1 ：

多项式时间归约

- 定理：若 $Q_1 \leq_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \leq_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

由于$Q_2 \in \mathrm{P}$ 知， Q_2有多项式时间算法，设为 A_2。

可如下构造 Q_1的多项式时间算法 A_1 ：

(1) 对任意 Q_1的输入 x，在多项式时间内计算 $f(x)$；

(2) 对 $f(x)$，利用算法 A_2 计算结果$A_2\left(f(x)\right)$；

(3) 对Q_1的输入 x，返回结果 $A_2\left(f(x)\right)$。

多项式时间归约

- 定理：若 $Q_1 \le_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \le_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

由于$Q_2 \in \mathrm{P}$ 知， Q_2有多项式时间算法，设为 A_2。

可如下构造 Q_1的多项式时间算法 A_1 ：

(1) 对任意 Q_1的输入 x，在多项式时间内计算 $f(x)$；

(2) 对 $f(x)$，利用算法 A_2 计算结果$A_2\big(f(x)\big)$；

(3) 对Q_1的输入 x，返回结果 $A_2\big(f(x)\big)$。

由多项式时间归约的定义知，以上算法A_1是正确的。

多项式时间归约

- 定理：若 $Q_1 \leq_P Q_2$ 且 $Q_2 \in \mathrm{P}$，则 $Q_1 \in \mathrm{P}$。

证明： 由$Q_1 \leq_P Q_2$知，存在多项式时间归约 f， 使得

对Q_1的任意实便 x，$f(x)$是 Q_2的一个实例。

由于$Q_2 \in \mathrm{P}$ 知， Q_2有多项式时间算法，设为 A_2。

可如下构造 Q_1的多项式时间算法 A_1 ：

(1) 对任意 Q_1的输入 x，在多项式时间内计算 $f(x)$；

(2) 对 $f(x)$，利用算法 A_2 计算结果$A_2(f(x))$；

(3) 对Q_1的输入 x，返回结果 $A_2(f(x))$。

由多项式时间归约的定义知，以上算法A_1是正确的。

由于第(1)、(2)步均是多项式时间的，因此算法A_1是多项式时间算法。

多项式归约的传递性

- 引理：给定问题 Q_1, Q_2 和 Q_3，若 $Q_1 \le_P Q_2$ 且$Q_2 \le_P Q_3$ ，则

$$Q_1 \le_P Q_3。$$

证明：由$Q_1 \le_P Q_2$ 知，存在一个从Q_1 到 Q_2 的多项式时间规约 f_1。
因此，对 Q_1 的任意实例 x， $|f_1(x)|$ 关于 $|x|$ 是多项式的。
同理，由 $Q_2 \le_P Q_3$ 知， 存在一个从Q_2 到 Q_3 的多项式时间规约 f_2，
因此， $|f_2(f_1(x))|$ 关于 $|x|$ 是多项式的。
可证： $f_2 \circ f_1$ 是从Q_1 到 Q_3 的多项式时间规约。（略）

NP-完全理论

- 多项式时间归约
- NP-完全类
- NP-完全问题

NP-完全类（NPC）

- 定义：如果一个判定问题 Q 满足以下两个条件：

  (1) $Q \in$ NP，且

  NPC ⊆ NP

  (2) 对于任意 $Q' \in$ NP，均有$Q' \leq_P Q$，

  NP 中的所有问题都不比 Q 难

则称 Q 是 NP完全（NP-complete）问题。

所有的NP-完全判定问题的集合构成NP完全类，记为NPC。

NP-完全类（NPC）

- 定义：如果一个判定问题 Q 满足以下两个条件：

  (1) $Q \in \mathrm{NP}$，且

  NPC ⊆ NP

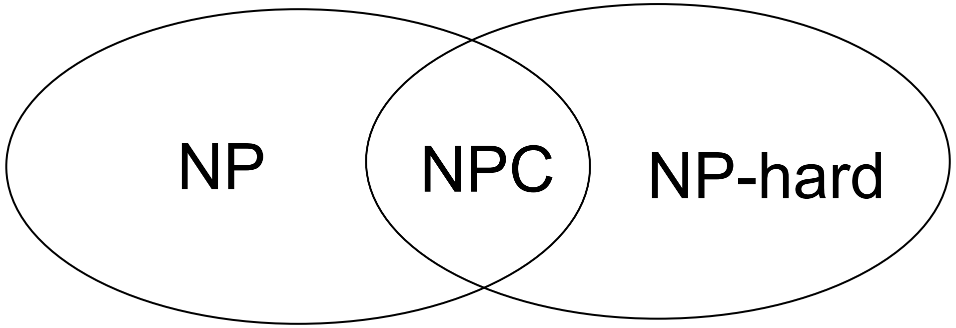
  (2) 对于任意 $Q' \in \mathrm{NP}$，均有$Q' \leq_P Q$，

  NP 中的所有问题都不比 Q 难

则称 Q 是 NP完全（NP-complete）问题。

所有的NP-完全判定问题的集合构成NP完全类，记为NPC。

- 如果一个判定问题 Q 满足条件(2)，则称 Q 是NP难（NP-hard）判定问题



NP-完全类（NPC）

- 定义：如果一个判定问题 Q 满足以下两个条件：

  (1) $Q \in$ NP，且

  NPC ⊆ NP

  (2) 对于任意 $Q' \in$ NP，均有$Q' \leq_P Q$，

  NP 中的所有问题都不比 Q 难

则称 Q 是 NP完全（NP-complete）问题。

所有的NP-完全判定问题的集合构成NP完全类，记为NPC。

- NPC包含了NP中所有最难的问题
  - 对任意 $L_1, L_2 \in$ NPC，有 $L_1 \leq_P L_2$ 且 $L_2 \leq_P L_1$

NP完全的性质

- 定理： 如果存在一个 NP 完全问题是多项式时间可解的，则所有NP问题均为多项式可解。

NP完全的性质

- 定理： 如果存在一个 NP 完全问题是多项式时间可解的，则所有NP问题均为多项式可解。

证明：设 $Q \in \mathrm{NPC}$，且 Q 有多项式求解算法。

NP完全的性质

- 定理： 如果存在一个 NP 完全问题是多项式时间可解的，则所有NP问题均为多项式可解。

证明：设 $Q \in$NPC，且 Q 有多项式求解算法。

由 $Q \in$NPC 知，对任意 $Q' \in$ NP，有 $Q' \leq_P Q$。

NP完全的性质

- 定理： 如果存在一个 NP 完全问题是多项式时间可解的，则所有NP问题均为多项式可解。

证明：设 Q ∈NPC，且 Q 有多项式求解算法。

由 Q ∈NPC 知，对任意 $Q' \in$ NP，有 $Q' \leq_P Q$。

因此， Q' 有多项式算法。

得，所有NP问题均为多项式可解。

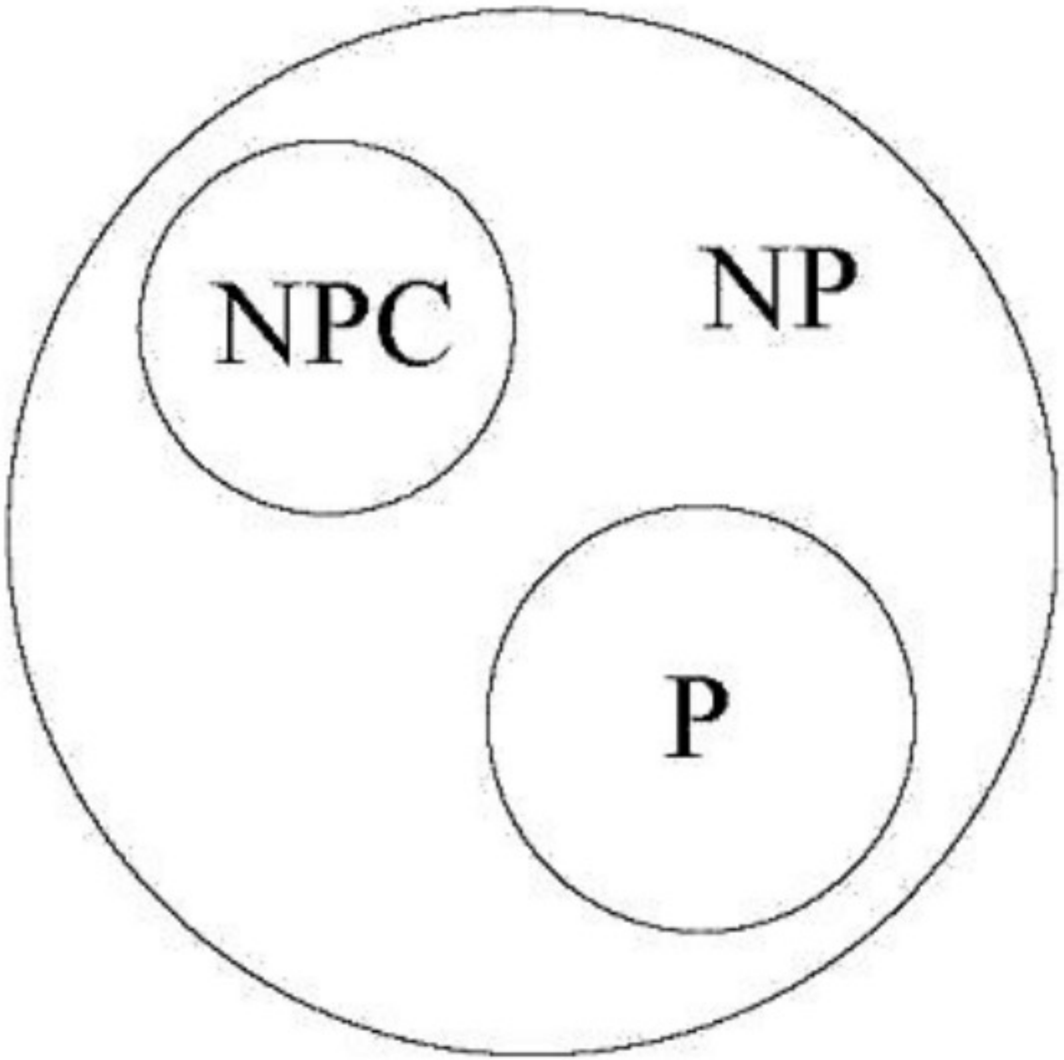
NP完全的性质

- 定理： 如果存在一个 NP 完全问题是多项式时间可解的，则所有NP问题均为多项式可解。

- 如果存在一个 NP 完全问题是多项式时间可解的，则有 P = NP。
- 目前，仍未有一个NP完全问题有多项式时间算法。
- 要么所有NP完全问题均为多项式时间可解，要么均不能多项式时间可解，两者只有一个成立。

P、NP、NPC的关系

- $P \subseteq NP$
- $NPC \subseteq NP$
- 如果 $P \cap NPC \neq \emptyset$，则 $P = NP$
- $P=NP?$
  - Open problem !
  - 公认： $P \neq NP$



提纲

- 背景介绍
- 判定问题与优化问题
- P问题与多项式时间算法
- NP问题与多项式时间验证
- NP完全问题与多项式时间归约
- **NP完全问题证明**

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in \mathrm{NP}$，且

  (2) 对于任意 $Q' \in \mathrm{NP}$，均有$Q' \leq_P Q$，

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in \mathrm{NP}$，且　（步骤 1）

  (2) 对于任意 $Q' \in \mathrm{NP}$，均有$Q' \le_P Q$,（步骤 2）

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

- 如何证明一个问题 Q 是NP-完全问题？

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in \mathrm{NP}$，且　（步骤 1）

  (2) 对于任意 $Q' \in \mathrm{NP}$，均有$Q' \leq_P Q$,(步骤 2)

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

- 如何证明一个问题 Q 是NP-完全问题？

（步骤 2）$\forall Q' \in \mathrm{NP} \xrightarrow{\text{多项式时间归约 } f_{Q' \to Q}\ ?} Q$

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in \text{NP}$，且　（步骤 1）

  (2) 对于任意 $Q' \in \text{NP}$，均有$Q' \leq_P Q$,（步骤 2）

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

- 如何证明一个问题 Q 是NP-完全问题？

（步骤 2）$\forall Q' \in \text{NP} \xrightarrow{\text{多项式时间归约 } f_{Q' \to Q}\ ?} Q$

引理：给定问题 Q_1, Q_2和Q_3，若$Q_1 \leq_P Q_2$ 且$Q_2 \leq_P Q_3$ ，则$Q_1 \leq_P Q_3$。

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in \mathrm{NP}$，且 （步骤 1）

  (2) 对于任意 $Q' \in \mathrm{NP}$，均有$Q' \leq_P Q$,（步骤 2）

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

- 如何证明一个问题 Q 是NP-完全问题？

（步骤 2）

$$
\begin{tikzcd}
\forall Q' \in \mathrm{NP} \arrow[rr, "\text{多项式时间归约 } f_{Q'\to Q}\text{ ?}"] \arrow[dr, "\text{多项式时间归约 } f^1_{Q'\to Q''}"'] & & L \\
& \exists Q'' \in \mathrm{NPC} \arrow[ur, "\text{多项式时间归约 } f^2_{Q''\to Q}"'] &
\end{tikzcd}
$$

引理：给定问题 Q_1, Q_2和Q_3，若$Q_1 \leq_P Q_2$ 且$Q_2 \leq_P Q_3$ ，则$Q_1 \leq_P Q_3$。

NP完全问题

- 如果一个判定问题 Q 满足

  (1) $Q \in$ NP，且　（步骤 1）

  (2) 对于任意 $Q' \in$ NP，均有$Q' \leq_P Q$,(步骤 2)

则称 Q 是 NP-完全问题。

所有的NP-完全判定问题的集合构成NP-完全类，记为NPC。

- 如何证明一个问题 Q 是NP-完全问题？

  两步（上界与下界）

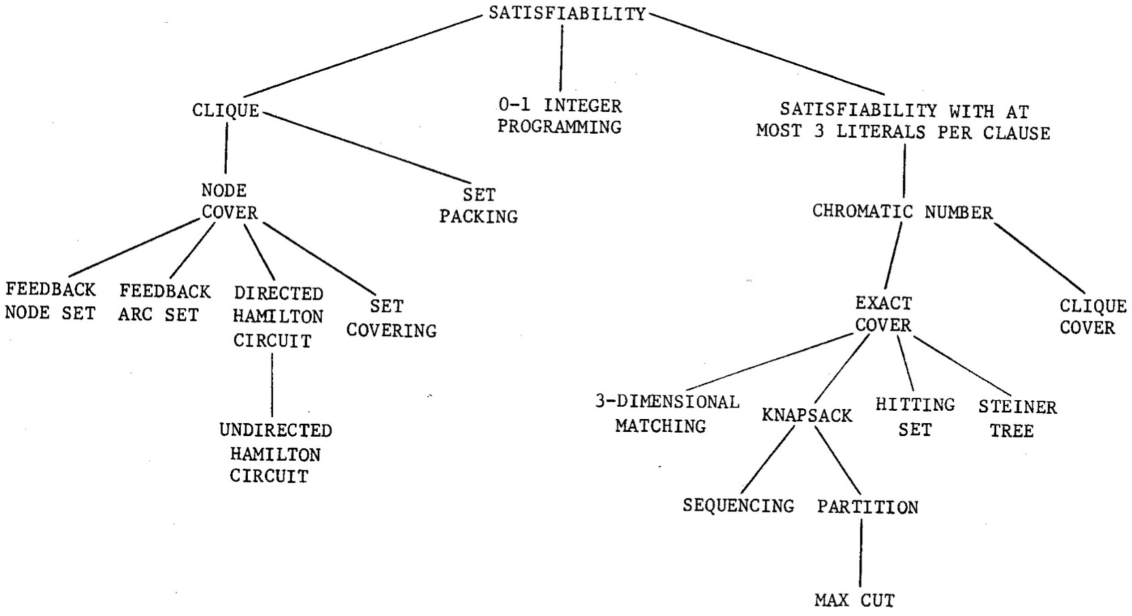
  (1) 上界： $Q \in$ NP （多项式时间可验证）

  (2) 下界：找到一个NP完全问题 Q_0，使得$Q_0 \leq_P Q$

第一个NP完全问题

- 布尔公式的可满足性问题（SAT）
  - Cook-Levin定理（20世纪 70年代由S.A.Cook、L.A. Levin分别独立证明）
- Karp's 21 NP-complete problems

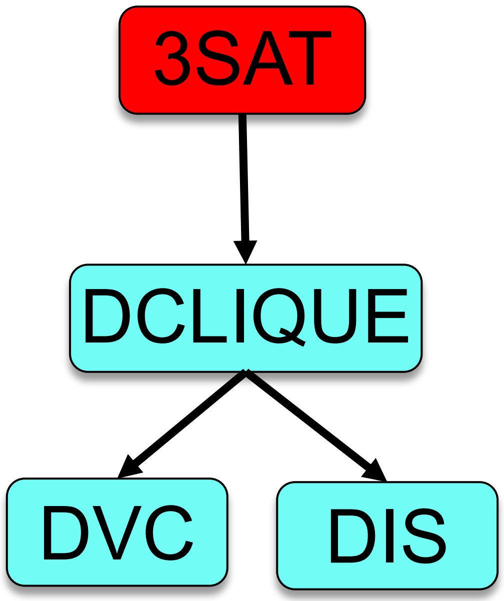
Stephen A. Cook　　Leonid Levin　　Richard M. Karp



- $3SAT \in NPC$
  - $SAT \leq_P 3SAT$

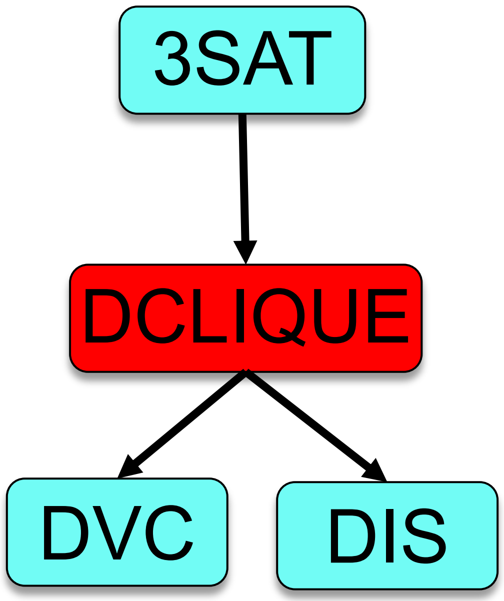
NP-完全问题证明

- 3-CNF 可满足性问题（3SAT）（已知）
- 团问题（DCLIQUE）
- 顶点覆盖问题（DVC）
- 独立集问题（DIS）



NP-完全问题证明

- 3-CNF 可满足性问题（3SAT）（已知）
- 团问题（DCLIQUE）
- 顶点覆盖问题（DVC）
- 独立集问题（DIS）



团问题

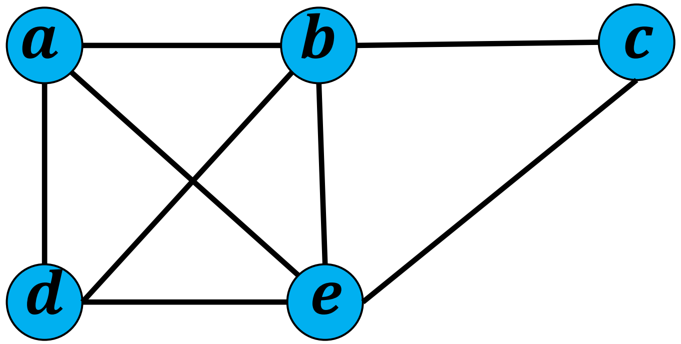
- 定义（团）：无向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}'\subseteq\boldsymbol{V}$，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u},\boldsymbol{v}\in\boldsymbol{V}'$，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u},\boldsymbol{v})\in\boldsymbol{E}$。
  - 一个团是图$\boldsymbol{G}$ 的一个完全子图

团问题

- 定义（团）：无向图 $G=(V,E)$ 的团是一个顶点子集 $V'\subseteq V$，满足 V' 中每一对顶点 $u,v\in V'$，在 G 中都有一条边 $(u,v)\in E$。
  - 一个团是图 G 的一个完全子图
- 团的规模：包含的顶点的个数

团问题

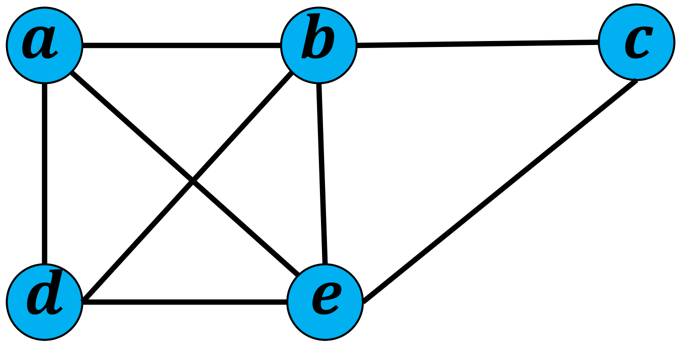
- 定义（团）：无向图 $G=(V,E)$ 的团是一个顶点子集 $V'\subseteq V$，满足 V' 中每一对顶点 $u,v\in V'$，在 G 中都有一条边$(u,v)\in E$。
  - 一个团是图G 的一个完全子图
- 团的规模：包含的顶点的个数



G_0

团问题

- 定义（团）：无向图 $G=(V,E)$ 的团是一个顶点子集 $V' \subseteq V$ ，满足 V' 中每一对顶点 $u,v \in V'$ ，在 G 中都有一条边 $(u,v) \in E$ 。
  - 一个团是图 G 的一个完全子图
- 团的规模：包含的顶点的个数

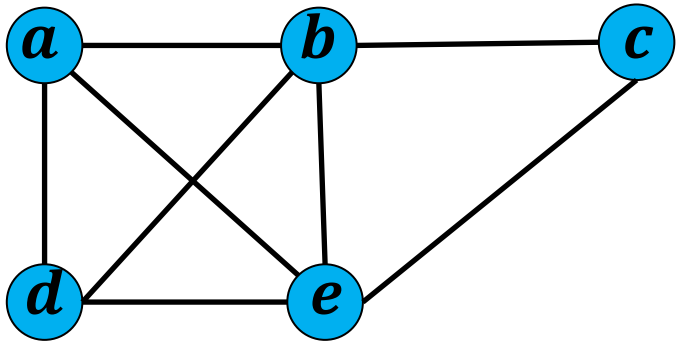


G_0

例：G_0 的规模为 1 的团：$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}$

团问题

- 定义（团）：无向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}' \subseteq \boldsymbol{V}$ ，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u},\boldsymbol{v} \in \boldsymbol{V}'$ ，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u},\boldsymbol{v}) \in \boldsymbol{E}$ 。
  - 一个团是图$\boldsymbol{G}$ 的一个完全子图
- 团的规模：包含的顶点的个数
  - 一个顶点即为一个大小为1的团

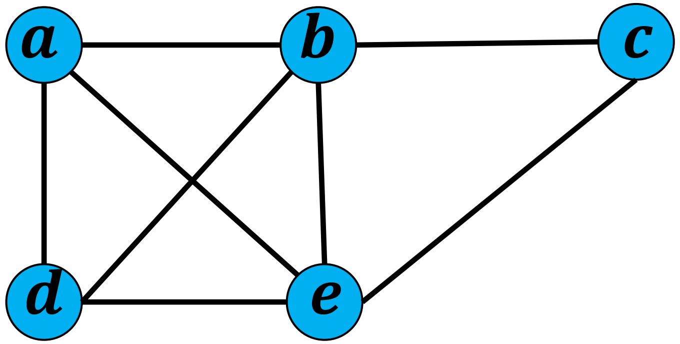


$\boldsymbol{G_0}$

例：$\boldsymbol{G_0}$ 的规模为 1 的团：$\{a\},\{b\},\{c\},\{d\},\{e\}$

团问题

- 定义（团）：无向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}'\subseteq\boldsymbol{V}$，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u},\boldsymbol{v}\in\boldsymbol{V}'$，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u},\boldsymbol{v})\in\boldsymbol{E}$。
  - 一个团是图$\boldsymbol{G}$ 的一个完全子图
- 团的规模：包含的顶点的个数
  - 一个顶点即为一个大小为1的团



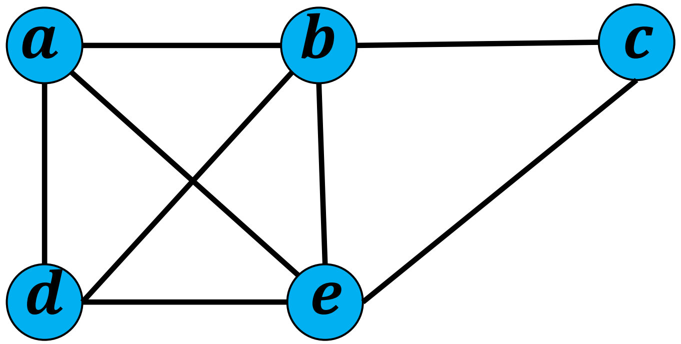
$\boldsymbol{G_0}$

例：$\boldsymbol{G_0}$ 的规模为 1 的团：$\{a\},\{b\},\{c\},\{d\},\{e\}$

规模为 2 的团：$\{a,b\},\{a,d\},\{a,e\},\{b,d\},\{d,e\},\{b,c\},\{b,e\},\{c,e\}$

团问题

- 定义（团）：无向图 $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}' \subseteq \boldsymbol{V}$ ，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{V}'$ ，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E}$ 。
  - 一个团是图$\boldsymbol{G}$ 的一个完全子图
- 团的规模：包含的顶点的个数
  - 一个顶点即为一个大小为1的团
  - 一条边即为一个大小为2的团



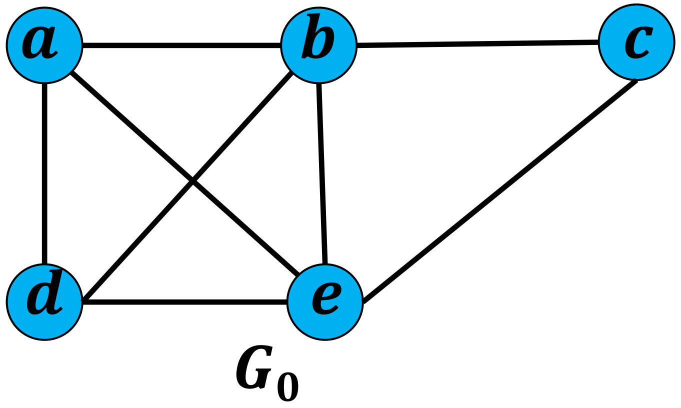
$\boldsymbol{G_0}$

例：$\boldsymbol{G_0}$ 的规模为 1 的团：$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}$

规模为 2 的团：$\{a, b\}, \{a, d\}, \{a, e\}, \{b, d\}, \{d, e\}, \{b, c\}, \{b, e\}, \{c, e\}$

团问题

- 定义（团）：无向图 $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}' \subseteq \boldsymbol{V}$，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{V}'$，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E}$。
  - 一个团是图$\boldsymbol{G}$ 的一个完全子图
- 团的规模：包含的顶点的个数
  - 一个顶点即为一个大小为1的团
  - 一条边即为一个大小为2的团



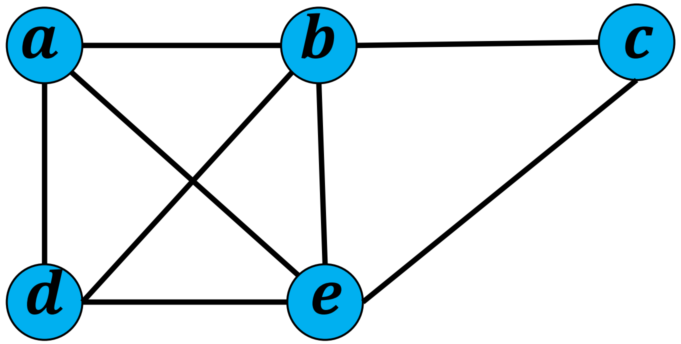
例：$\boldsymbol{G}_0$ 的规模为 1 的团：$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}$

规模为 2 的团：$\{a, b\}, \{a, d\}, \{a, e\}, \{b, d\}, \{d, e\}, \{b, c\}, \{b, e\}, \{c, e\}$

规模为 3 的团：$\{a, d, b\}, \{a, d, e\}, \{a, b, e\}, \{b, d, e\}, \{b, c, e\}$

团问题

- 定义（团）：无向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 的团是一个顶点子集 $\boldsymbol{V}'\subseteq\boldsymbol{V}$，满足 $\boldsymbol{V}'$ 中每一对顶点 $\boldsymbol{u},\boldsymbol{v}\in\boldsymbol{V}'$，在 $\boldsymbol{G}$ 中都有一条边$(\boldsymbol{u},\boldsymbol{v})\in\boldsymbol{E}$。
  - 一个团是图$\boldsymbol{G}$的一个完全子图
- 团的规模：包含的顶点的个数
  - 一个顶点即为一个大小为1的团
  - 一条边即为一个大小为2的团



$\boldsymbol{G_0}$

例：$\boldsymbol{G_0}$ 的规模为 1 的团：$\{a\},\{b\},\{c\},\{d\},\{e\}$

规模为 2 的团：$\{a,b\},\{a,d\},\{a,e\},\{b,d\},\{d,e\},\{b,c\},\{b,e\},\{c,e\}$

规模为 3 的团：$\{a,d,b\},\{a,d,e\},\{a,b,e\},\{b,d,e\},\{b,c,e\}$

规模为 4 的团：$\{a,\ b,\ c,\ e\}$

团问题

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？

团问题

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？

团问题

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为k的团？
- 定理：团问题是NP问题（CLIQUE$\in$ NPC）。

DCLIQUE ∈ NPC

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为k的团？
- 定理：团问题是NP问题（CLIQUE∈ NPC）。

证明思想：两步（上界与下界）

(1) 上界：CLIQUE∈ NP

(2) 下界：找到一个NP完全问题 L，使得 $L \leq_P$CLIQUE

上界证明：DCLIQUE ∈ NP

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？
- DCLIQUE ∈ NP

上界证明：DCLIQUE ∈ NP

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？
- DCLIQUE ∈ NP

证明： 对任意无向图 $G=(V,E)$，给定 G 的顶点集 $V'\subseteq V$，其中 $|V'|=k$，下面证明可以在多项式时间内验证 V' 是 G 的一个团。

上界证明：DCLIQUE ∈ NP

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？
- DCLIQUE ∈ NP

证明： 对任意无向图 $G=(V,E)$，给定 G 的顶点集 $V' \subseteq V$，其中 $|V'|=k$，下面证明可以在多项式时间内验证 V' 是 G 的一个团。

即验证：对 每对顶点 $u,v \in V'$，$u \neq v$，有 $(u,v) \in E$。

上界证明：DCLIQUE ∈ NP

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？
- DCLIQUE ∈ NP

证明： 对任意无向图 $G = (V, E)$，给定 G 的顶点集 $V' \subseteq V$，其中 $|V'| = k$，下面证明可以在多项式时间内验证 V' 是 G 的一个团。

即验证：对 每对顶点 $u, v \in V'$，$u \neq v$，有 $(u, v) \in E$。

显然，当把 G 表示为邻接矩阵时，以上验证可在 $O(|V|^2)$时间内完成。

上界证明：DCLIQUE ∈ NP

- 团问题（DCLIQUE）
  - 输入：无向图 G，整数 k
  - 问题： G中是否有一个规模为 k 的团？
- DCLIQUE ∈ NP

证明： 对任意无向图 $G=(V,E)$，给定 G 的顶点集 $V' \subseteq V$，其中 $|V'|=k$，下面证明可以在多项式时间内验证 V' 是 G 的一个团。

即验证：对 每对顶点 $u,v \in V'$，$u \neq v$，有 $(u,v) \in E$。

显然，当把 G 表示为邻接矩阵时，以上验证可在 $O(|V|^2)$时间内完成。

因此，团问题是NP问题。

下界证明： 3SAT $\leq_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明： 已知 3SAT 是NP完全问题，下面证明 3SAT $\leq_P$DCLIQUE。

下界证明：3SAT $\leq_P$DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明： 已知 3SAT 是NP完全问题，下面证明 3SAT $\leq_P$DCLIQUE。

给定 3SAT 的任意实例 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ ，其中 φ 是定义在变量集 $X = \{x_1, x_2, \ldots, x_m\}$ 的3-合取范式，即对任意 $i \in [1, n]$,

$$C_i = l_1^i \vee l_2^i \vee l_3^i \text{，} \ l_j^i \in X \text{ 或 } \neg l_j^i \in X \text{，}$$

下界证明： 3SAT $\le_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明： 已知 3SAT 是NP完全问题，下面证明 3SAT $\le_P$DCLIQUE。

给定 3SAT 的任意实例 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ ，其中 φ 是定义在变量集 $X = \{x_1, x_2, \ldots, x_m\}$ 的3-合取范式，即对任意 $i \in [1, n]$,

$$C_i = l_1^i \vee l_2^i \vee l_3^i \text{，} \quad l_j^i \in X \text{ 或 } \neg l_j^i \in X \text{，}$$

构造DCLIQUE的实例 (G, k) ，满足

下界证明：3SAT $\le_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明： 已知 3SAT 是NP完全问题，下面证明 3SAT $\le_P$DCLIQUE。

给定 3SAT 的任意实例 $\varphi = C_1 \land C_2 \land \ldots \land C_n$，其中 φ 是定义在变量集 $X = \{x_1, x_2, \ldots, x_m\}$ 的3-合取范式，即对任意 $i \in [1, n]$,

$$C_i = l_1^i \lor l_2^i \lor l_3^i \text{，} \quad l_j^i \in X \text{ 或 } \neg l_j^i \in X \text{，}$$

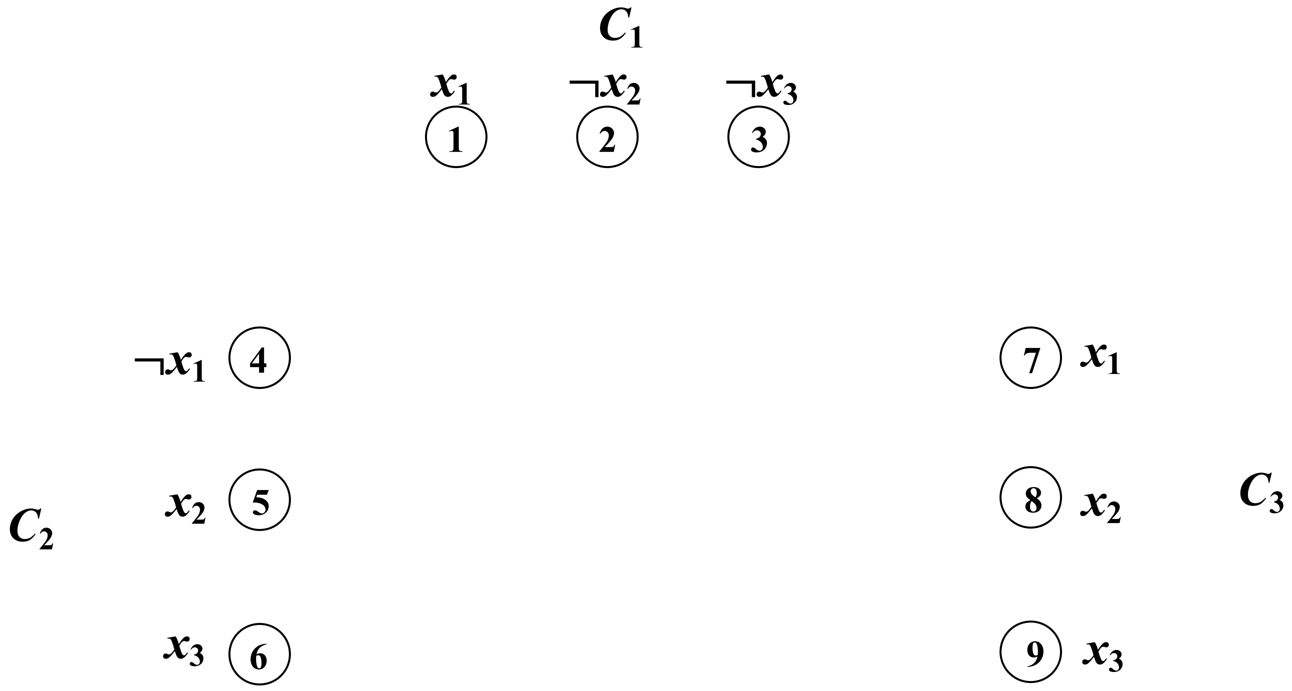
构造DCLIQUE的实例 (G, k)，满足

φ 是可满足的当且仅当 G 中有一个大小为 k 的团。

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

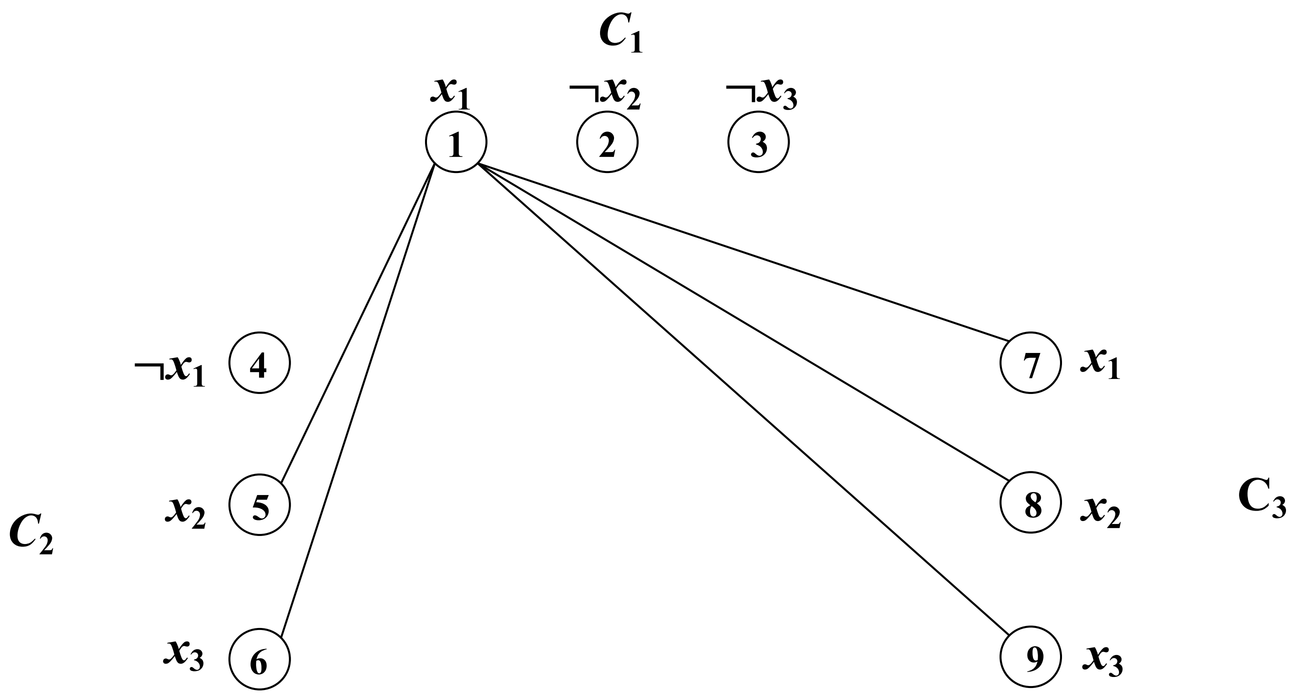
$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.



举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

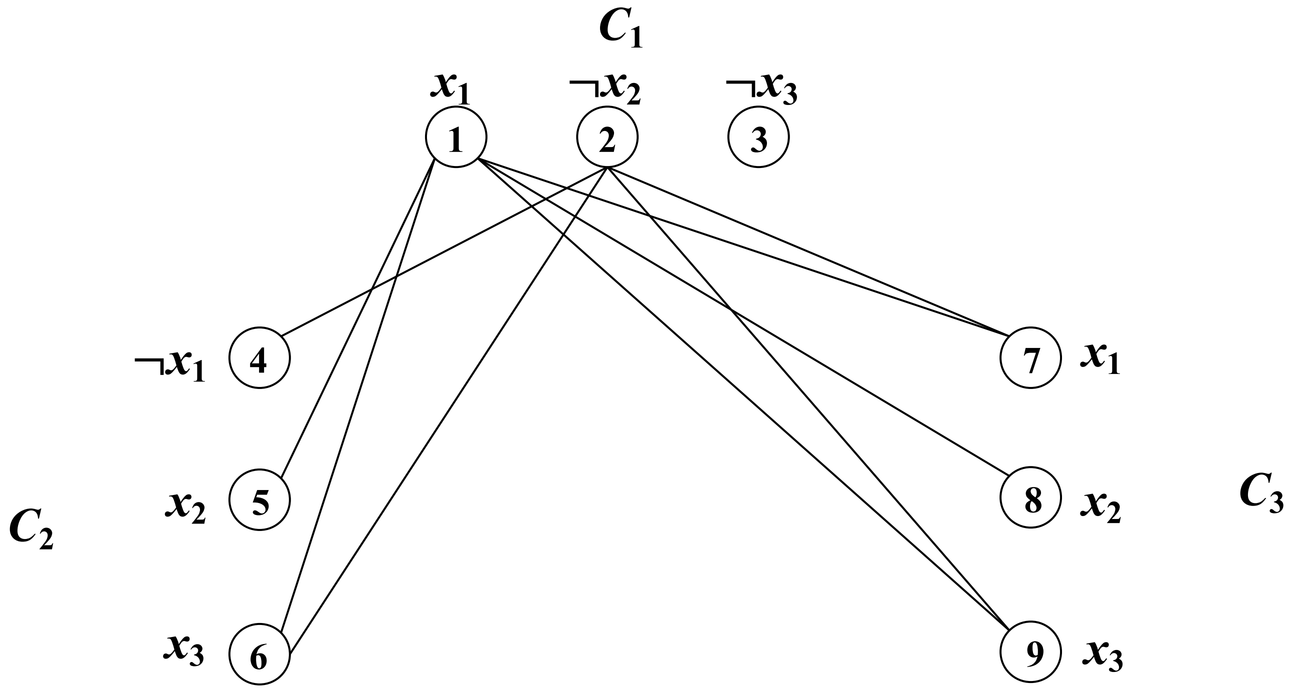
$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.



举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

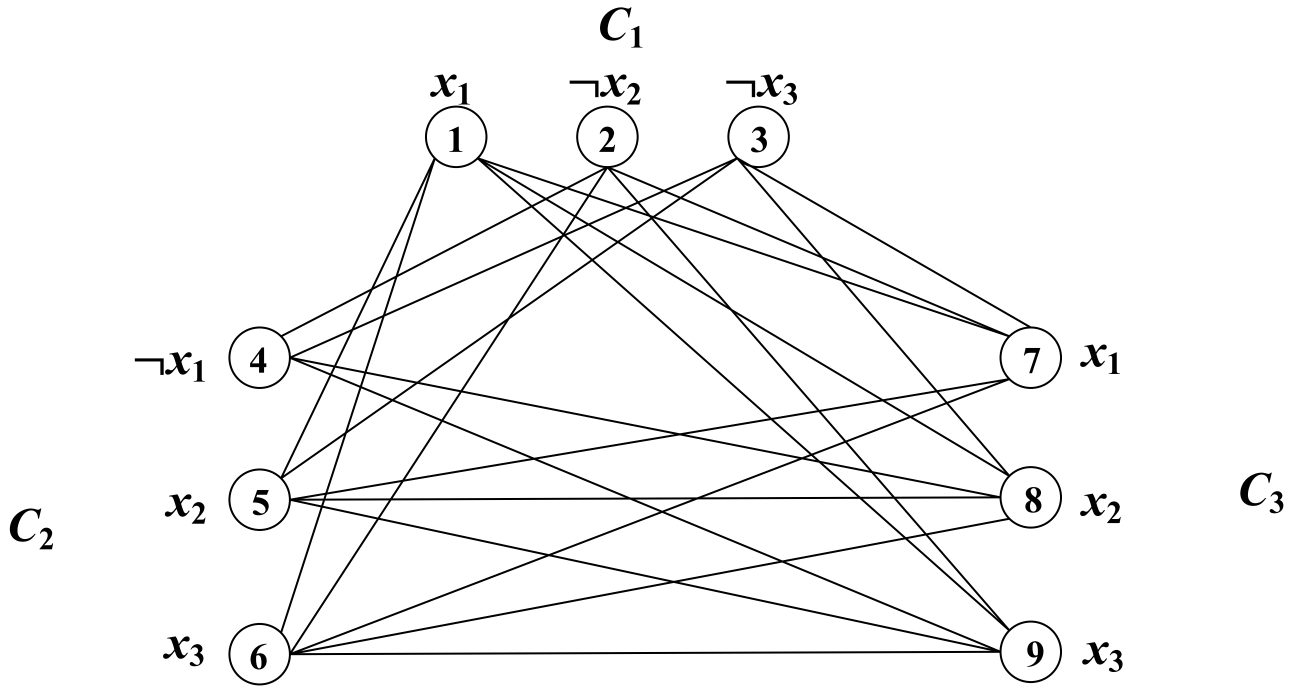
$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.



举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

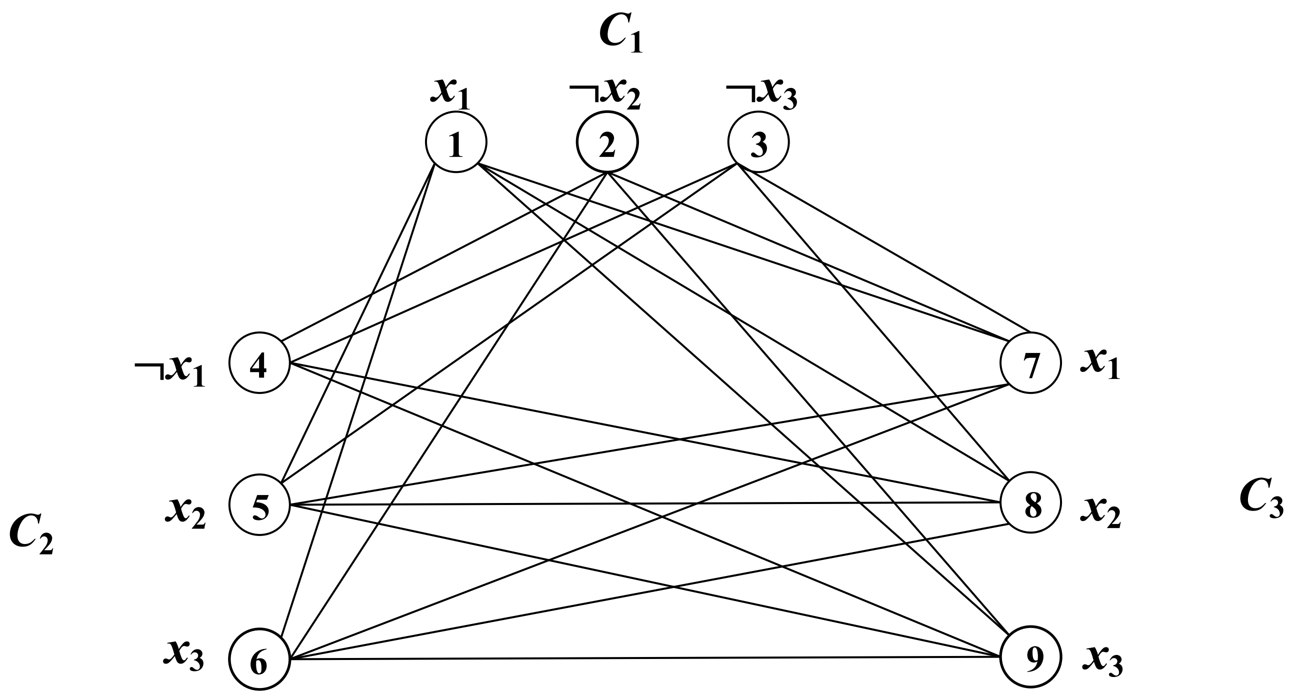
$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.



举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.

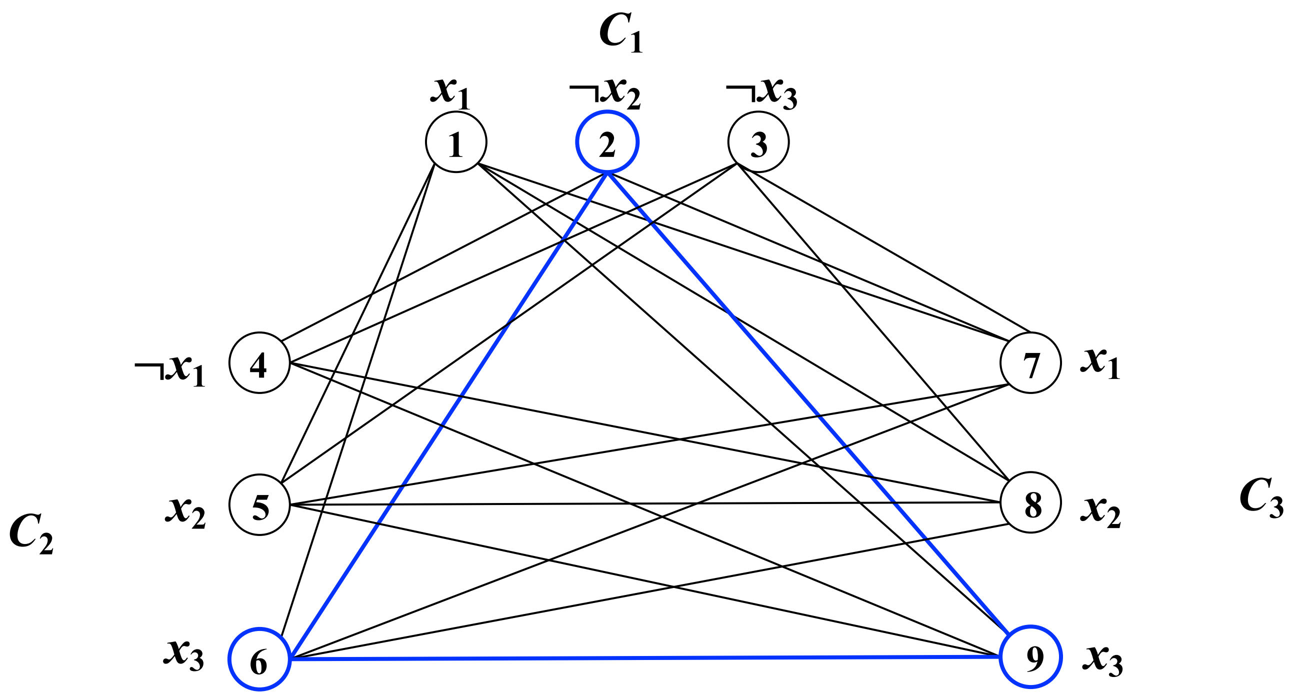


- 赋值 $(x_1 = 0, x_2 = 0, x_3 = 1)$ 使得 ϕ为真

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.

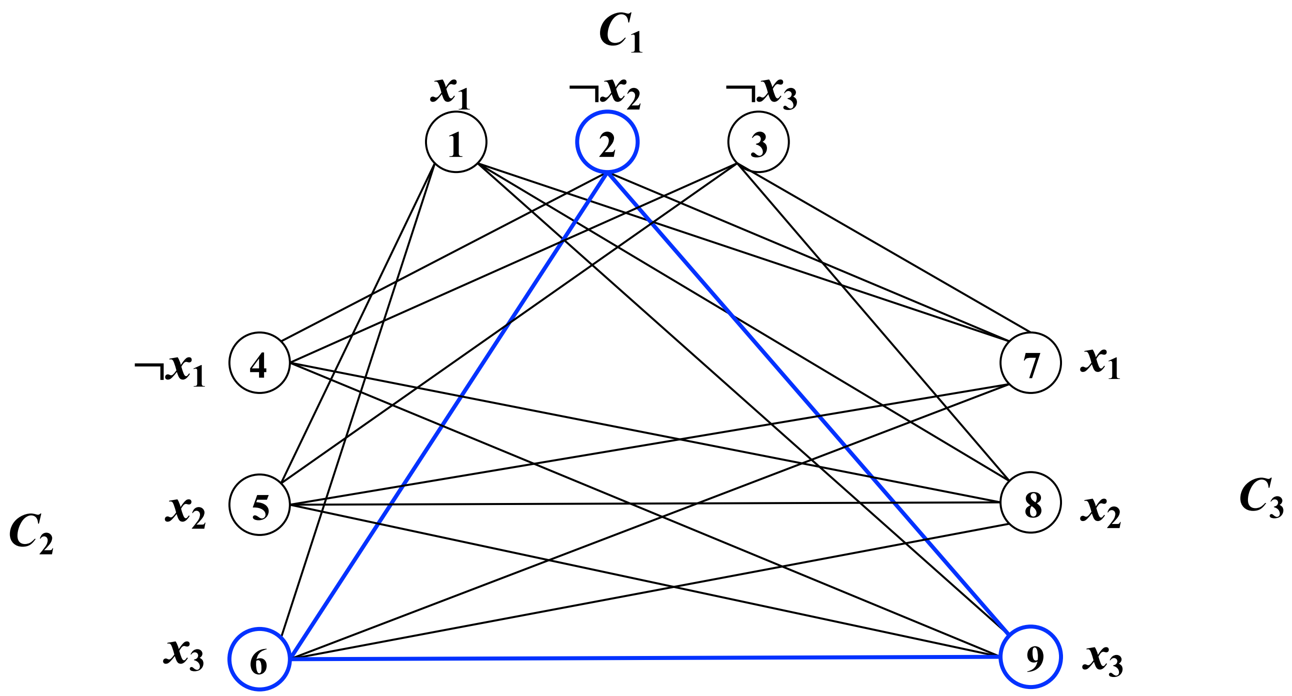


- 赋值 $(x_1 = 0, x_2 = 0, x_3 = 1)$ 使得 ϕ 为真

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.

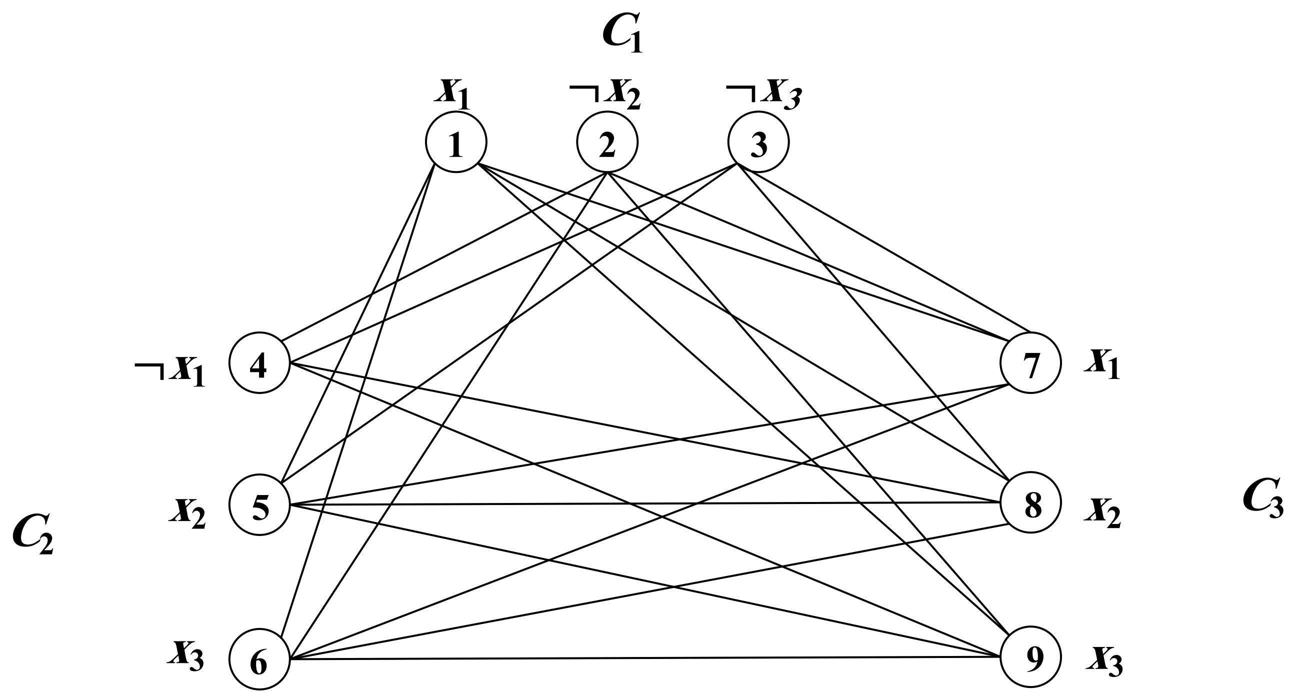


- 赋值 (x_1 = 0, x_2 =0, x_3 =1) 使得 ϕ为真
- 节点2, 6, 9 构成一个大小为3的团

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.

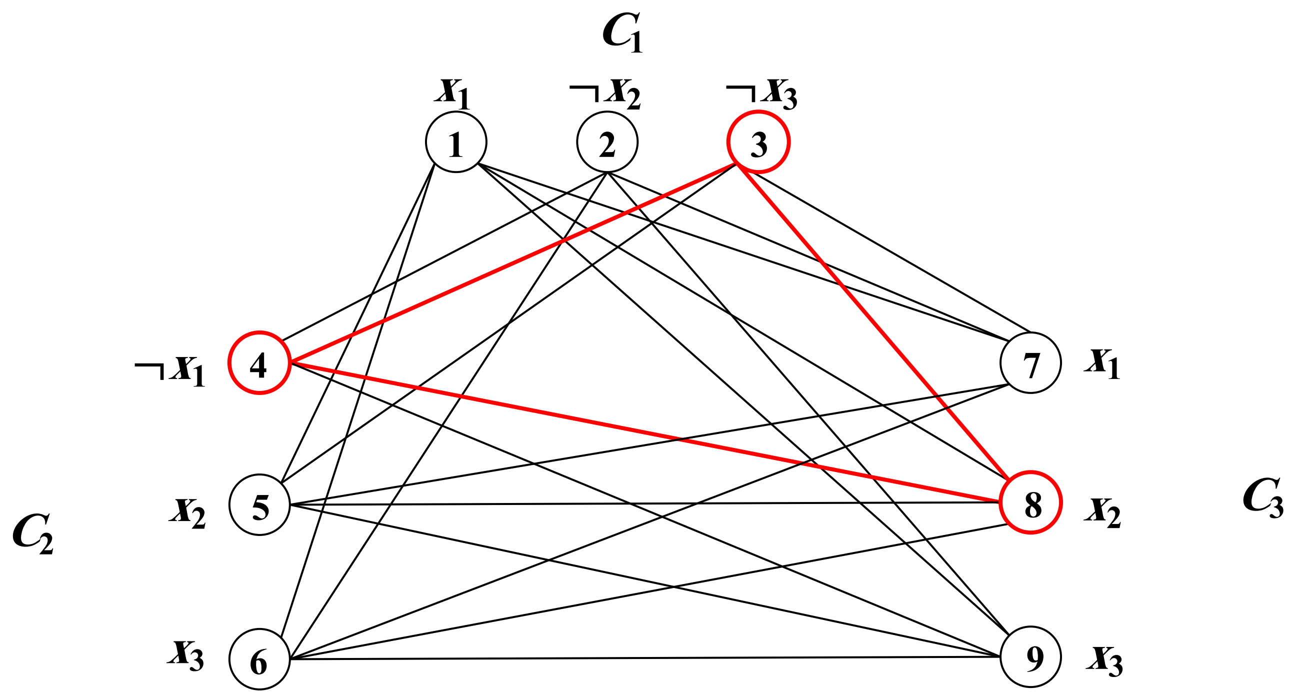


- 赋值 $(x_1 = 0, x_2 = 1, x_3 = 0)$ 使得 ϕ为真

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.

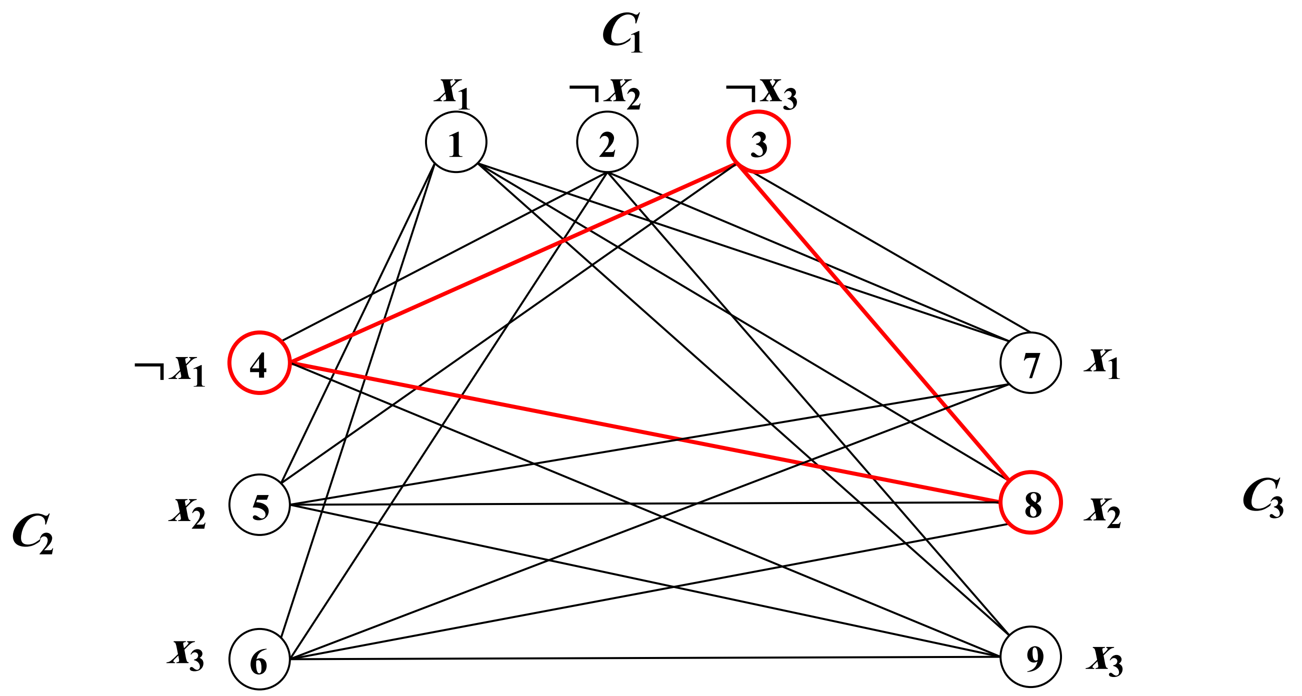


- 赋值 $(x_1 = 0, x_2 = 1, x_3 = 0)$ 使得 ϕ为真

举例说明

例如，令 $\varphi = C_1 \wedge C_2 \wedge C_3$，其中

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$， $C_2 = \neg x_1 \vee x_2 \vee x_3$，$C_3 = x_1 \vee x_2 \vee x_3$.



- 赋值 $(x_1 = 0, x_2 = 1, x_3 = 0)$ 使得 ϕ为真
- 节点 3, 4, 8 构成一个大小为 3 的团

下界证明：3SAT $\le_P$DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明： 已知 3SAT 是NP完全问题，下面证明 3SAT $\le_P$DCLIQUE。

给定 3SAT 的任意实例 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$，其中 φ 是定义在变量集 $X = \{x_1, x_2, \ldots, x_m\}$ 的3-合取范式，即对任意 $i \in [1, n]$,

$$C_i = l_1^i \vee l_2^i \vee l_3^i\,,\quad l_j^i \in X \text{ 或 } \neg l_j^i \in X\,,$$

构造DCLIQUE的实例 (G, k)，满足

φ 是可满足的当且仅当 G 中有一个大小为k的团。

下界证明：3SAT $\leq_P$DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G = (V, E)$:

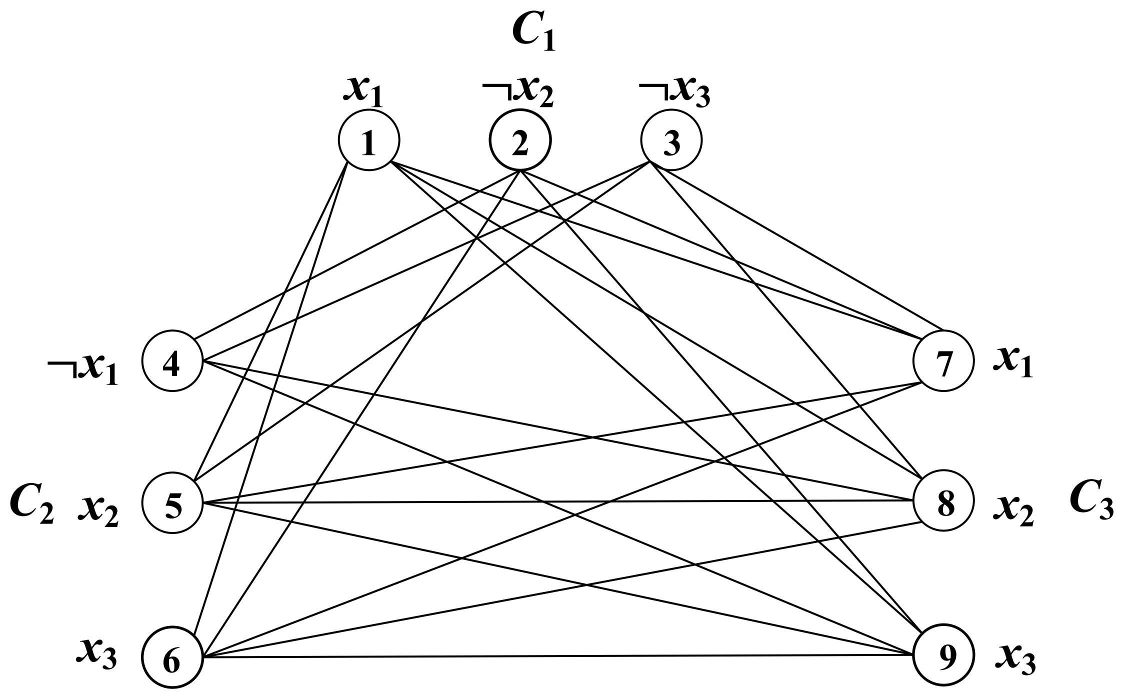
下界证明： 3SAT $\leq_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G = (V, E)$:

(a) 构造顶点集 V：

(b) 构造边集 E：

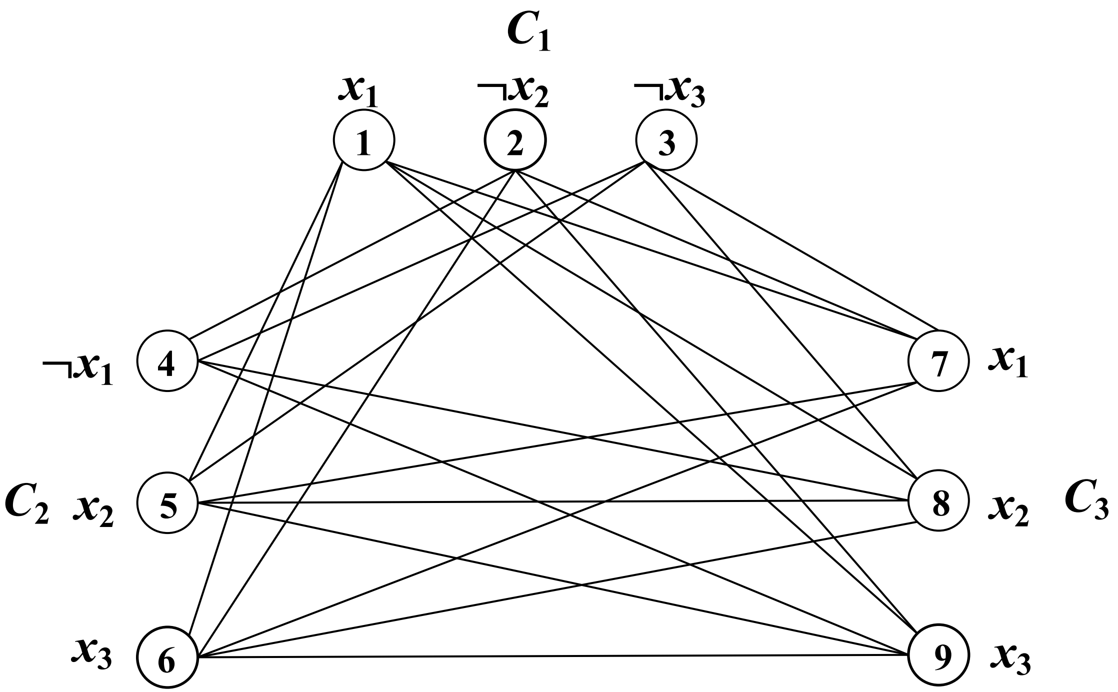


下界证明：3SAT $\leq_P$DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G = (V, E)$:

(a) 构造顶点集 V： 对每一个子句 $C_i = l_1^i \vee l_2^i \vee l_3^i$， V包含3个顶点 v_1^i, v_2^i, v_3^i，分别对应 l_1^i, l_2^i, l_3^i。



(b) 构造边集 E：

下界证明： 3SAT $\leq_P$ DCLIQUE

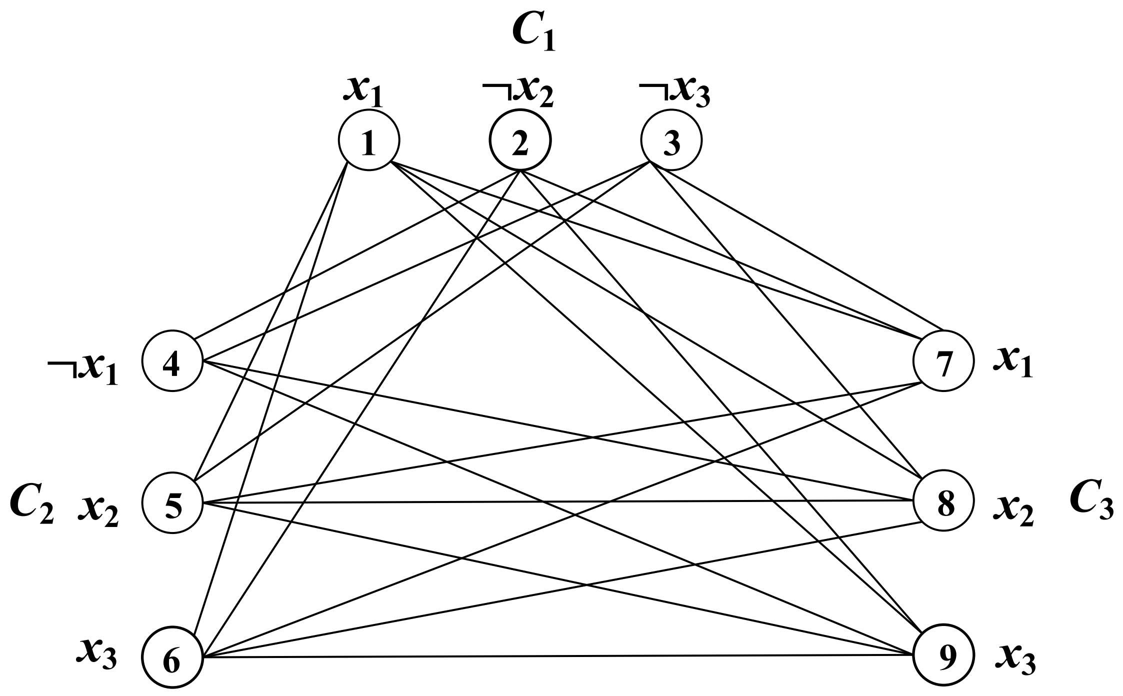
- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G = (V, E)$:

(a) 构造顶点集 V： 对每一个子句 $C_i = l_1^i \vee l_2^i \vee l_3^i$， V包含3个顶点 v_1^i, v_2^i, v_3^i，分别对应 l_1^i, l_2^i, l_3^i。

因此， V中共有 $3n$ 个顶点。

(b) 构造边集 E：



下界证明： 3SAT $\le_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G=(V,E)$:

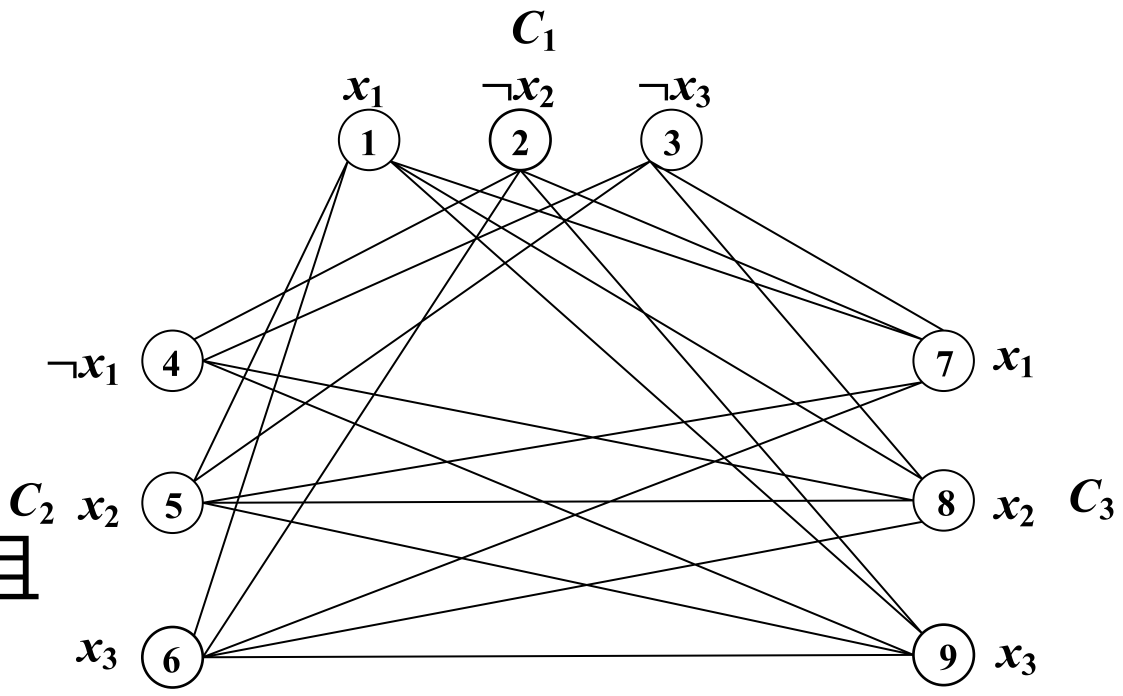
(a) 构造顶点集 V： 对每一个子句 $C_i = l_1^i \vee l_2^i \vee l_3^i$， V包含3个顶点 v_1^i, v_2^i, v_3^i，分别对应 l_1^i, l_2^i, l_3^i。

因此， V中共有 $3n$ 个顶点。

(b) 构造边集 E：

顶点 v_j^i 与 $v_{j'}^{i'}$间有一条边当且仅当

- ✓ l_j^i 与 $l_{j'}^{i'}$位于不同的子句，即 $i \neq i'$，且
- ✓ l_j^i 不是 $l_{j'}^{i'}$的否，即 $l_j^i \neq \neg l_{j'}^{i'}$。



下界证明：3SAT $\le_P$ DCLIQUE

- 定理：团问题是NP完全问题（DCLIQUE∈ NPC）。

证明（续）： 如下构造DCLIQUE实例 $G = (V, E)$:

(a) 构造顶点集 V： 对每一个子句 $C_i = l_1^i \vee l_2^i \vee l_3^i$， V包含3个顶点 v_1^i, v_2^i, v_3^i，分别对应 l_1^i, l_2^i, l_3^i。

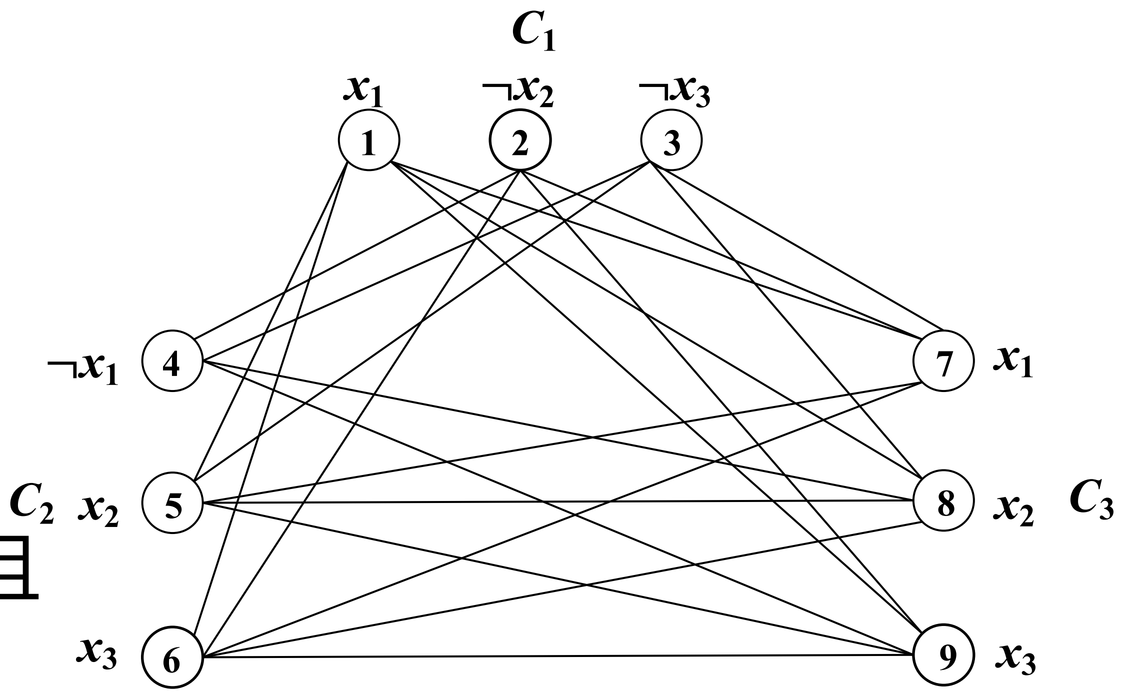
因此， V中共有 $3n$ 个顶点。

(b) 构造边集 E：

顶点 v_j^i 与 $v_{j'}^{i'}$间有一条边当且仅当

- ✓ l_j^i 与 $l_{j'}^{i'}$位于不同的子句，即 $i \neq i'$，且
- ✓ l_j^i 不是 $l_{j'}^{i'}$的否，即 $l_j^i \neq \neg l_{j'}^{i'}$。

(c) 令 $k = n$。



下界证明： 3SAT $\le_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \land C_2 \land \ldots \land C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

下界证明：3SAT $\leq_P$ DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

$(\Longrightarrow)$ 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。

下界证明：3SAT $\le_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

$(\Longrightarrow)$ 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。因此，每个子句 C_i 中至少存在一个文字的真值为1。

下界证明：3SAT $\leq_P$ DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

$(\Longrightarrow)$ 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。

因此，每个子句 C_i 中至少存在一个文字的真值为1。

不失一般性，假设 C_i中子句 $l_{j_i}^{i}$ 的真值为 1（$i = 1, \ldots, n, j_i \in \{1,2,3\}$）。

下界证明：3SAT $\le_P$ DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

($\Longrightarrow$) 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。因此，每个子句 C_i 中至少存在一个文字的真值为1。

不失一般性，假设 C_i中子句 $l^i_{j_i}$ 的真值为 1（$i = 1, \ldots, n, j_i \in \{1,2,3\}$）。

$\{l^i_{j_i} | \, i = 1, \ldots, n\}$ 满足：每个文字来源于不同的子句，且任意两个文字都不是一个变量与该变量的否。

下界证明： 3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

($\Longrightarrow$) 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。因此，每个子句 C_i 中至少存在一个文字的真值为1。

不失一般性，假设 C_i中子句 $l_{j_i}^i$ 的真值为 1 （$i = 1, \ldots, n, j_i \in \{1,2,3\}$）。

$\{l_{j_i}^i | \ i = 1, \ldots, n\}$ 满足：每个文字来源于不同的子句，且任意两个文字都不是一个变量与该变量的否。

因此，对应顶点子集 $\{v_{j_i}^i | \ i = 1, \ldots, n\}$中，任意两个顶点间必有一条边。

下界证明：3SAT $\leq_P$ DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$ 是可满足的 当且仅当 G 中有一个大小为 k 的团。

($\Longrightarrow$) 假设 φ 是可满足，则存在一个赋值 μ 使得所有子句 C_i 的真值为1。

因此，每个子句 C_i 中至少存在一个文字的真值为1。

不失一般性，假设 C_i中子句 $l_{j_i}^i$ 的真值为 1（$i = 1, \ldots, n, j_i \in \{1,2,3\}$）。

$\{l_{j_i}^i | \ i = 1, \ldots, n\}$ 满足：每个文字来源于不同的子句，且任意两个文字都不是一个变量与该变量的否。

因此，对应顶点子集 $\{v_{j_i}^i | \ i = 1, \ldots, n\}$中，任意两个顶点间必有一条边。

故$\{v_{j_i}^i | \ i = 1, \ldots, n\}$ 构成一个大小为 $k = n$ 的团。

下界证明： 3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

($\Leftarrow$) 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

下界证明：3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

$(\Longleftarrow)$ 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

不妨设团为 $\{v_{j_i}^i \mid i = 1, \ldots, n\}$。

下界证明： 3SAT $\le_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

$(\Longleftarrow)$ 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

不妨设团为 $\{v_{j_i}^i | \ i = 1, \ldots, n\}$。

由于团中任意两个顶点间有边，因此对应的 $\{l_{j_i}^i | \ i = 1, \ldots, n\}$ 中任意两个文字都不是一个变量及该变量的否。

下界证明： 3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

$(\Longleftarrow)$ 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

不妨设团为 $\{v_{j_i}^i | \ i = 1, \ldots, n\}$。

由于团中任意两个顶点间有边，因此对应的 $\{l_{j_i}^i | \ i = 1, \ldots, n\}$ 中任意两个文字都不是一个变量及该变量的否。

如下定义真值赋值 μ：

(1) 若 $l_{j_i}^i = x_k \in X$，则$\mu(x_k) = 1$；若 $l_{j_i}^i = \neg x_k$，则$\mu(x_k) = 0$；

下界证明：3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

$(\Longleftarrow)$ 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

不妨设团为 $\{v_{j_i}^i \mid i = 1, \ldots, n\}$。

由于团中任意两个顶点间有边，因此对应的 $\{l_{j_i}^i \mid i = 1, \ldots, n\}$ 中任意两个文字都不是一个变量及该变量的否。

如下定义真值赋值 μ：

(1) 若 $l_{j_i}^i = x_k \in X$，则$\mu(x_k) = 1$；若 $l_{j_i}^i = \neg x_k$，则$\mu(x_k) = 0$；

(2) 对未在 $\{l_{j_i}^i \mid i = 1, \ldots, n\}$ 中出现的变量，可任意赋值。

下界证明：3SAT $\leq_P$DCLIQUE

证明（续）：下面证明 $\varphi = C_1 \land C_2 \land \dots \land C_n$是可满足的当且仅当$G$中有一个大小为 $k = n$的团。

($\Leftarrow$) 假设G中有一个大小为 $k = n$ 的团，则团中每个顶点来自于不同的子句。

不妨设团为 $\{v_{j_i}^i \mid i = 1, \dots, n\}$。

由于团中任意两个顶点间有边，因此对应的 $\{l_{j_i}^i \mid i = 1, \dots, n\}$ 中任意两个文字都不是一个变量及该变量的否。

如下定义真值赋值 μ：

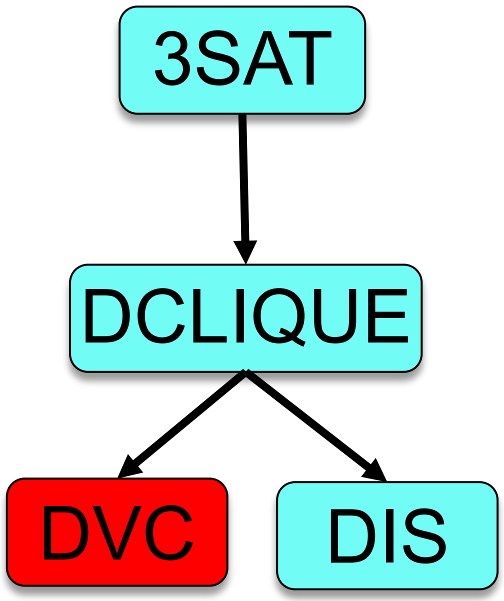
(1) 若 $l_{j_i}^i = x_k \in X$，则$\mu(x_k) = 1$；若 $l_{j_i}^i = \neg x_k$，则$\mu(x_k) = 0$；

(2) 对未在 $\{l_{j_i}^i \mid i = 1, \dots, n\}$ 中出现的变量，可任意赋值。

则 μ 使每个子句C_i都为真，因此 φ 是可满足的。

NP-完全问题证明

- 3-CNF 可满足性问题（3SAT）（已知）
- 团问题（DCLIQUE）
- 顶点覆盖问题（DVC）
- 独立集问题（DIS）



顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V'\subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V' \subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

  - V 是 G 的一个顶点覆盖

顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V'\subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

  - V 是 G 的一个顶点覆盖
- 顶点覆盖的规模：包含的顶点的个数

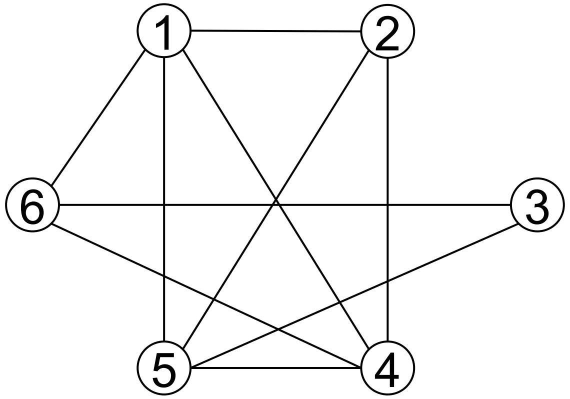
顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V' \subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

  - V 是 G 的一个顶点覆盖

- 顶点覆盖的规模：包含的顶点的个数



G

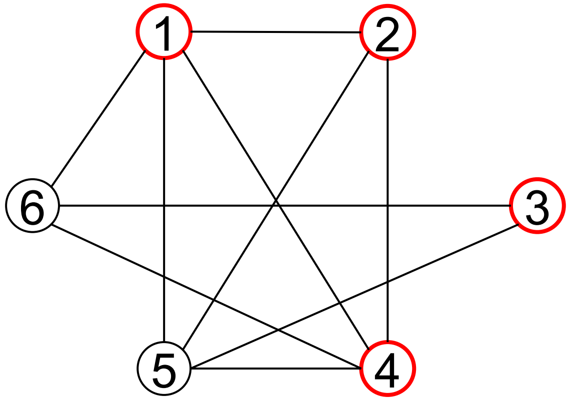
$\{1,2,3,4,5,6\}$ 是 G 的一个顶点覆盖。

顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G = (V, E)$，G的一个顶点覆盖为 G的顶点子集$V' \subseteq V$，使得

  对于G中任意一条边 $e = (u, v) \in E$，$u \in V'$ 或$v \in V'$。

  - V 是 G 的一个顶点覆盖
- 顶点覆盖的规模：包含的顶点的个数



G

$\{1, 2, 3, 4, 5, 6\}$ 是 G 的一个顶点覆盖。

$\{1, 2, 3, 4\}$ 是 G 的一个顶点覆盖。

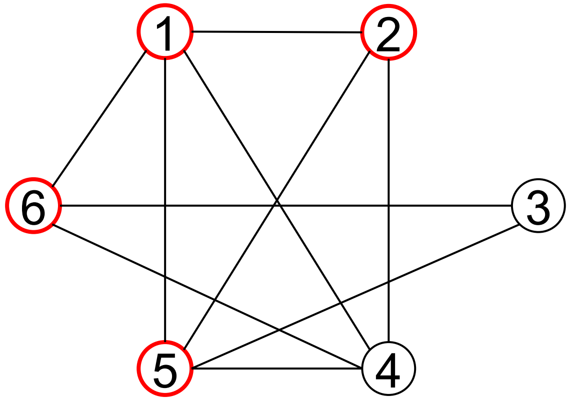
顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V'\subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

  - V 是 G 的一个顶点覆盖

- 顶点覆盖的规模：包含的顶点的个数



G

$\{1,2,3,4,5,6\}$ 是 G 的一个顶点覆盖。

$\{1,2,3,4\}$ 是 G 的一个顶点覆盖。

$\{1,2,5,6\}$ 是 G 的一个顶点覆盖。

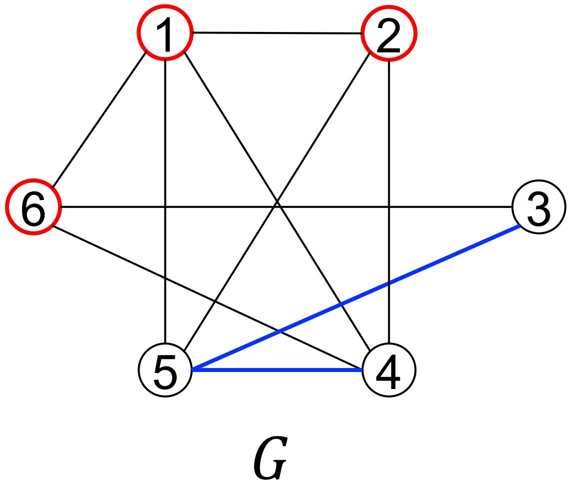
顶点覆盖问题

- 定义（顶点覆盖）给定无向图 $G=(V,E)$， G的一个顶点覆盖为 G的顶点子集$V' \subseteq V$，使得

  对于G中任意一条边 $e=(u,v)\in E$， $u\in V'$ 或$v\in V'$。

  - V 是 G 的一个顶点覆盖

- 顶点覆盖的规模：包含的顶点的个数



$\{1,2,3,4,5,6\}$ 是 G 的一个顶点覆盖。

$\{1,2,3,4\}$ 是 G 的一个顶点覆盖。

$\{1,2,5,6\}$ 是 G 的一个顶点覆盖。

$\{1,2,6\}$ 不是 G 的顶点覆盖。

顶点覆盖问题

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？

顶点覆盖问题

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？
- 定理：顶点覆盖问题是NP完全问题（ DVC ∈NPC）。

证明思想：两步（上界与下界）

(1) 上界： DVC ∈ NP (已证)

(2) 下界：找到一个NP完全问题 L，使得 $L \leq_P$ DVC

- DCLIQUE $\leq_P$ DVC

上界证明：DVC $\in$NP

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？
- DVC $\in$NP

上界证明：DVC ∈NP

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？
- DVC ∈NP

证明： 对任意无向图 $G=(V,E)$ 和顶点子集 $V'\subseteq V$，其中 $|V'|=k$，下面证明可以多项式时间内验证 V'是一个顶点覆盖，

上界证明：DVC ∈NP

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？

- DVC ∈NP

证明： 对任意无向图 $G=(V,E)$ 和顶点子集 $V'\subseteq V$，其中 $|V'|=k$，

下面证明可以多项式时间内验证 V'是一个顶点覆盖，

即验证：对每条边 $(u,v)\in E$， $u\in V'$ 或 $v\in V'$。

上界证明：DVC ∈NP

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？

- DVC ∈NP

证明： 对任意无向图 $G=(V,E)$ 和顶点子集 $V' \subseteq V$，其中 $|V'|=k$，下面证明可以多项式时间内验证 V'是一个顶点覆盖，

即验证：对每条边 $(u,v) \in E$， $u \in V'$ 或 $v \in V'$。

显然 ，当把G 表示为邻接矩阵时，以上验证可在 $O(|V|^2)$时间内完成。

上界证明：DVC ∈NP

- 顶点覆盖（DVC）问题
  - 输入：无向图 G，正整数 k
  - 问题： G 是否有一个大小为k 的顶点覆盖（即包含k 个顶点）？

- DVC ∈NP

证明： 对任意无向图 $G=(V,E)$ 和顶点子集 $V'\subseteq V$，其中 $|V'|=k$，下面证明可以多项式时间内验证 V'是一个顶点覆盖，

即验证：对每条边 $(u,v)\in E$， $u\in V'$ 或 $v\in V'$。

显然 ，当把G 表示为邻接矩阵时，以上验证可在 $O(|V|^2)$时间内完成。

因此，顶点覆盖问题是NP问题。

下界证明：DCLIQUE $\leq_P$ DVC

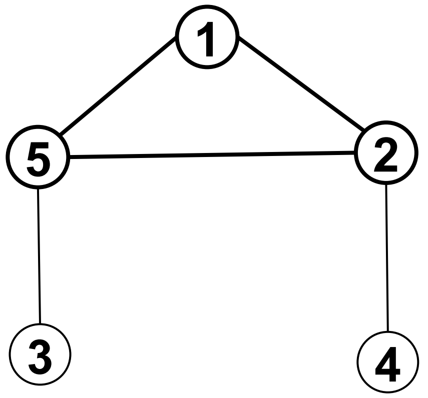
证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例 (G', k')，使得

G有一个大小为 k的团 当且仅当 G' 有一个大小为 k' 的顶点覆盖。

下界证明：DCLIQUE $\leq_P$ DVC

设 $G=(V,E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。

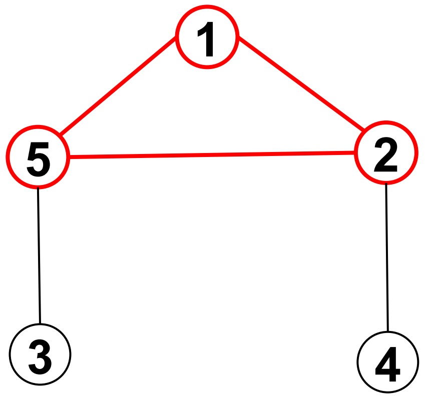


$G=(V,E)$

下界证明：DCLIQUE $\leq_P$ DVC

设 $G = (V, E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。



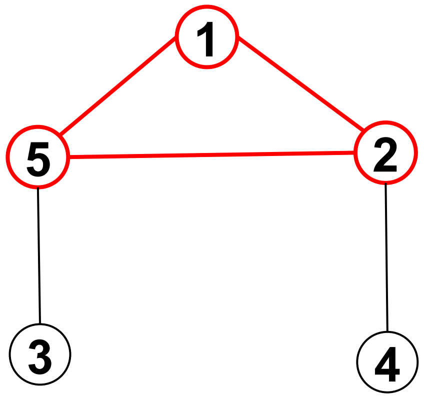
$G = (V, E)$

$\{1, 2, 5\}$
是 G 的一个团

下界证明：DCLIQUE $\leq_P$ DVC

设 $G = (V, E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。



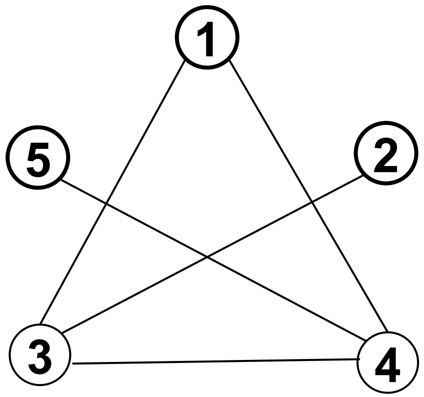
$G = (V, E)$

$\{1, 2, 5\}$

是 G 的一个团

构造补图

$(u, v) \in E' \Leftrightarrow (u, v) \notin E$

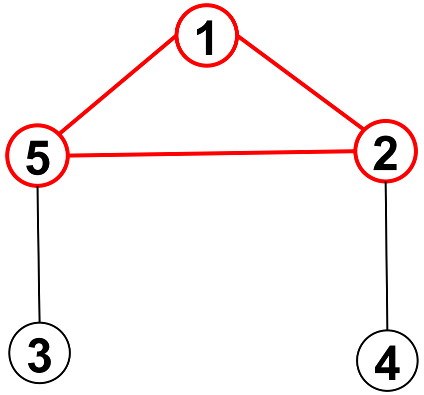


$\bar{G} = (V, E')$

下界证明：DCLIQUE $\leq_P$ DVC

设 $G = (V, E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。



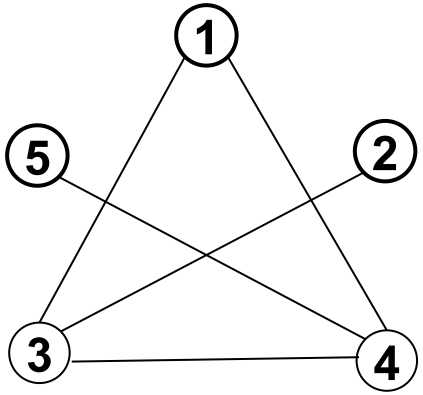
$G = (V, E)$

$\{1, 2, 5\}$

是 G 的一个团

构造补图

$(u, v) \in E' \Leftrightarrow (u, v) \notin E$



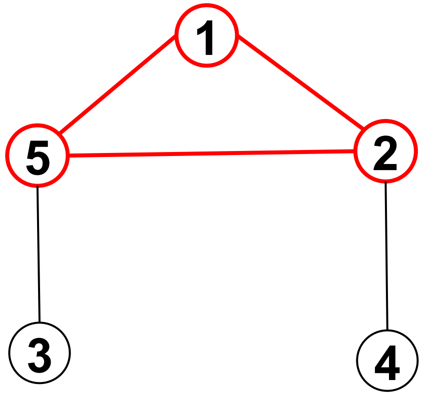
$\bar{G} = (V, E')$

- G 中顶点1, 2, 5之间的边都不在补图$\bar{G}$ 中出现

下界证明：DCLIQUE $\leq_P$ DVC

设 $G = (V, E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。

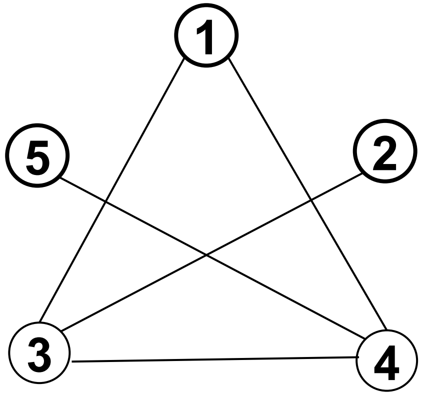


$G = (V, E)$

$\{1, 2, 5\}$
是 G 的一个团

构造补图

$(u, v) \in E' \Leftrightarrow (u, v) \notin E$



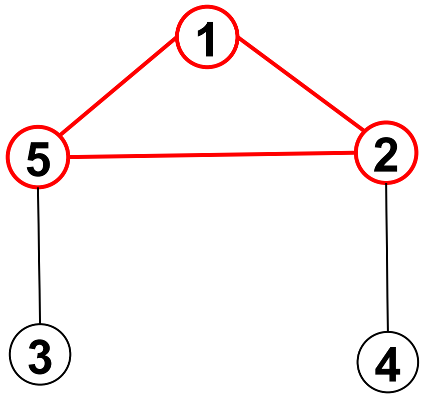
$\bar{G} = (V, E')$

- G 中顶点1, 2, 5之间的边都不在补图$\bar{G}$ 中出现
- 补图 $\bar{G}$ 中所有边只与顶点3或 4 相连

下界证明：DCLIQUE $\leq_P$ DVC

设 $G=(V,E)$是无向图，$V' \subseteq V$，则

- V' 是G 的一个团 当且仅当 对任意两个顶点 $v, v' \in V'$，有 $(v, v') \in E$；
- V' 是G 的一个顶点覆盖 当且仅当 对任意一条边$(v, v') \in E$，有 $v \in V'$ 或 $v' \in V'$。

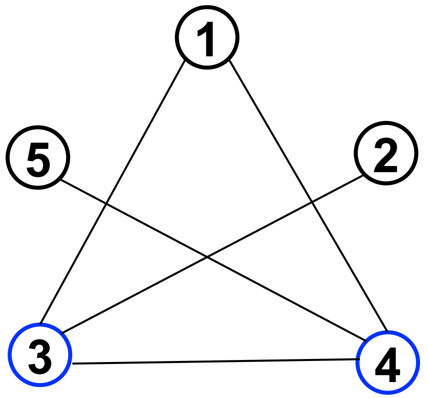


$G=(V,E)$

$\{1,2,5\}$
是 G 的一个团

构造补图

$(u, v) \in E' \Leftrightarrow (u, v) \notin E$



$\bar{G}=(V,E')$

$\{3,4\}$
是 $\bar{G}$ 的一个顶点覆盖

- G 中顶点$1,2,5$之间的边都不在补图$\bar{G}$ 中出现
- 补图 $\bar{G}$ 中所有边只与顶点3或 4 相连
- $\{3,4\}$构成补图$\bar{G}$ 的一个顶点覆盖

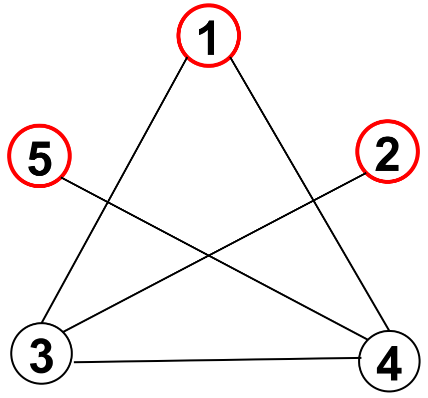
下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G,k)，其中 $G=(V,k)$是一个无向图，构造DVC问题的实例(G',k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G'=\bar{G}$， $k'=|V|-k$，其中 $\bar{G}=(V,E')$是 G的补图，满足

$(u,v)\in E'$ 当且仅当 $(u,v)\notin E$。



$\bar{G}=(V,E')$

下界证明：DCLIQUE $\leq_P$ DVC

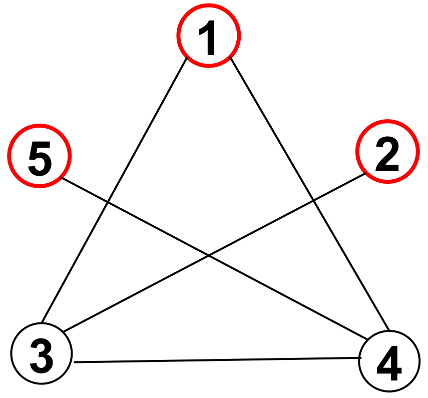
证明： 给定 DCLIQUE 问题的任意实例 (G,k)，其中 $G=(V,k)$是一个无向图，构造DVC问题的实例(G',k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G'=\bar{G}$， $k'=|V|-k$，其中 $\bar{G}=(V,E')$是 G的补图，满足

$$(u,v)\in E' \text{ 当且仅当 } (u,v)\notin E。$$

$(\Longrightarrow)$ 假设 G 有一个大小为 k 的团 V'，则 V' 中任意两个顶点之间都有一条边，而这些边在补图 $\bar{G}$ 不出现。



$\bar{G}=(V,E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G,k)，其中 $G=(V,k)$是一个无向图，构造DVC问题的实例(G',k')，使得

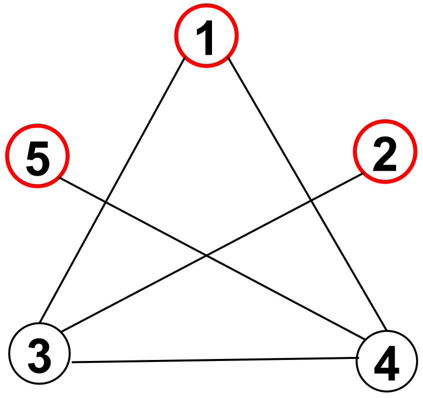
G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G'=\bar{G}$， $k'=|V|-k$，其中 $\bar{G}=(V,E')$是 G的补图，满足

$$(u,v)\in E' \text{ 当且仅当 } (u,v)\notin E。$$

$(\Longrightarrow)$ 假设 G 有一个大小为 k 的团 V'，则 V' 中任意两个顶点之间都有一条边，而这些边在补图 $\bar{G}$ 不出现。

因此，补图 $\bar{G}$ 中每条边都与 $V\backslash V'$中一个顶点相连。



$\bar{G}=(V,E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G,k)，其中 $G=(V,k)$是一个无向图，构造DVC问题的实例(G',k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

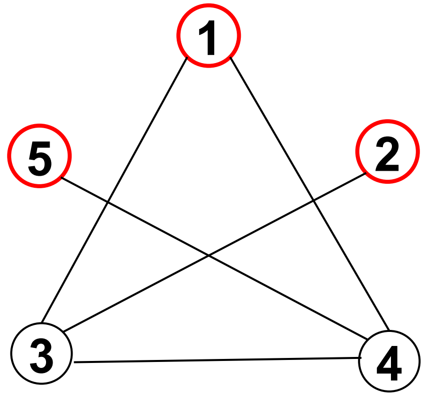
令 $G'=\bar{G}$， $k'=|V|-k$，其中 $\bar{G}=(V,E')$是 G的补图，满足

$$(u,v)\in E' \text{ 当且仅当 } (u,v)\notin E。$$

($\Longrightarrow$) 假设 G 有一个大小为 k 的团 V'，则 V' 中任意两个顶点之间都有一条边，而这些边在补图 $\bar{G}$ 不出现。

因此，补图 $\bar{G}$ 中每条边都与 $V\backslash V'$中一个顶点相连。

从而， $V\backslash V'$中节点构成补图 $\bar{G}$的一个大小



$\bar{G}=(V,E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例(G', k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

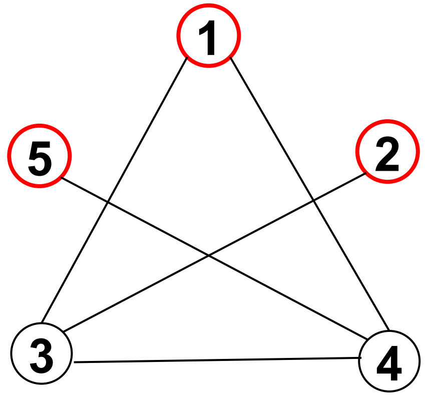
令 $G' = \bar{G}$， $k' = |V| - k$，其中 $\bar{G} = (V, E')$是 G的补图，满足

$$(u, v) \in E' \text{ 当且仅当 } (u, v) \notin E。$$

$(\Longrightarrow)$ 假设 G 有一个大小为 k 的团 V'，则 V' 中任意两个顶点之间都有一条边，而这些边在补图 $\bar{G}$ 不出现。

因此，补图 $\bar{G}$ 中每条边都与 $V \backslash V'$中一个顶点相连。

从而， $V \backslash V'$中节点构成补图 $\bar{G}$的一个大小为$|V| - k$的顶点覆盖。



$\bar{G} = (V, E')$

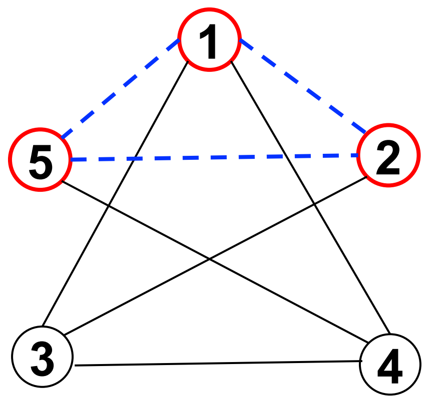
下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例(G', k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G' = \bar{G}$， $k' = |V| - k$，其中 $\bar{G} = (V, E')$是 G的补图，满足

$$(u, v) \in E' \text{ 当且仅当 } (u, v) \notin E。$$



$\bar{G} = (V, E')$

下界证明：DCLIQUE $\leq_P$ DVC

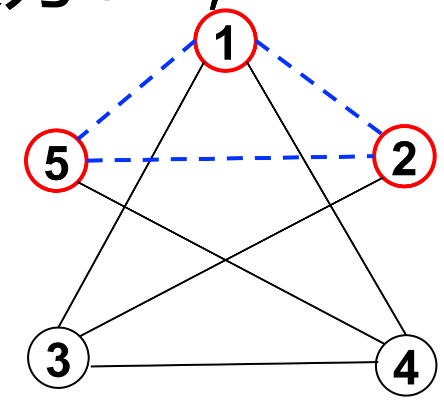
证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例(G', k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G' = \bar{G}$， $k' = |V| - k$，其中 $\bar{G} = (V, E')$是 G的补图，满足

$$(u, v) \in E' \text{ 当且仅当 } (u, v) \notin E。$$

$(\Longleftarrow)$ 假设补图 $\bar{G}$ 有一个大小为$|V| - k$的顶点覆盖，设为 V''，



$\bar{G} = (V, E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例(G', k')，使得

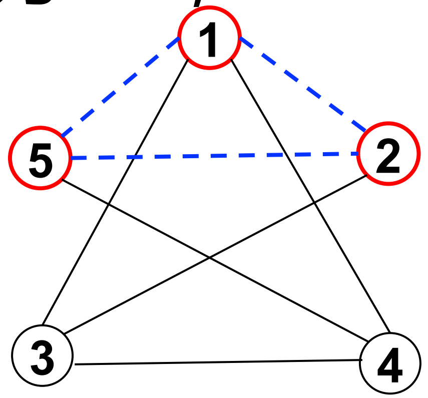
G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G' = \bar{G}$， $k' = |V| - k$，其中 $\bar{G} = (V, E')$是 G的补图，满足

$$(u, v) \in E' \text{ 当且仅当 } (u, v) \notin E。$$

$(\Longleftarrow)$ 假设补图 $\bar{G}$ 有一个大小为$|V| - k$的顶点覆盖，设为 V''，

则 $\bar{G}$中任意一条边都与 V'' 中一个顶点相连。



$\bar{G} = (V, E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G, k)，其中 $G = (V, k)$是一个无向图，构造DVC问题的实例(G', k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G' = \bar{G}$， $k' = |V| - k$，其中 $\bar{G} = (V, E')$是 G的补图，满足

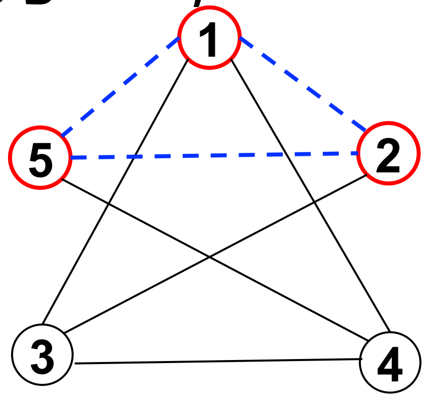
$$(u, v) \in E' \text{ 当且仅当 } (u, v) \notin E。$$

$(\Longleftarrow)$ 假设补图 $\bar{G}$ 有一个大小为$|V| - k$的顶点覆盖，设为 V''，

则 $\bar{G}$中任意一条边都与 V'' 中一个顶点相连。

因此，对任意的两个顶点 $u, v \in V \backslash V''$， $(u, v) \notin E'$，

则$(u, v) \in E$。



$\bar{G} = (V, E')$

下界证明：DCLIQUE $\leq_P$ DVC

证明： 给定 DCLIQUE 问题的任意实例 (G,k)，其中 $G=(V,k)$是一个无向图，构造DVC问题的实例(G',k')，使得

G有一个大小为 k的团 当且仅当 G'有一个大小为 k' 的顶点覆盖。

令 $G'=\bar{G}$， $k'=|V|-k$，其中 $\bar{G}=(V,E')$是 G的补图，满足

$$(u,v)\in E' \text{ 当且仅当 } (u,v)\notin E。$$

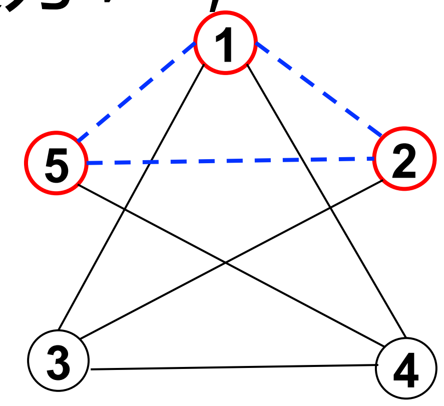
$(\Longleftarrow)$ 假设补图 $\bar{G}$ 有一个大小为$|V|-k$的顶点覆盖，设为 V''，

则 $\bar{G}$中任意一条边都与 V'' 中一个顶点相连。

因此，对任意的两个顶点 $u,v\in V\backslash V''$， $(u,v)\notin E'$，

则$(u,v)\in E$。

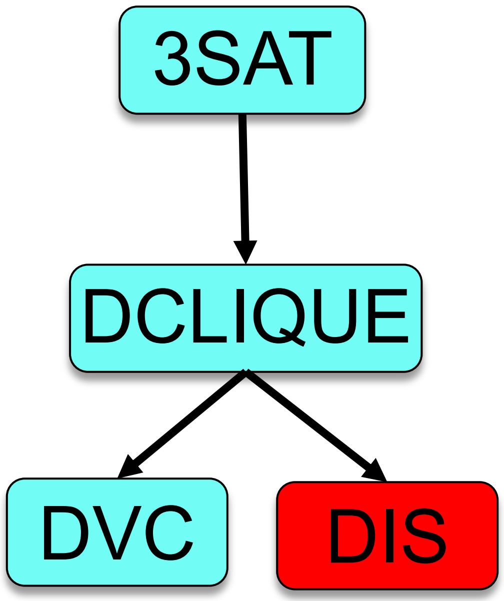
故 $V\backslash V''$ 构成 G 的一个大小为k的团。



$\bar{G}=(V,E')$

NP-完全问题证明

- 3-CNF 可满足性问题（3SAT）（已知）
- 团问题（DCLIQUE）
- 顶点覆盖问题（DVC）
- 独立集问题（DIS）



独立集问题

- 定义（独立集）给定无向图 $G=(V,E)$，G 的独立子集为一个顶点子集 $V'\subseteq V$ 满足：

V'中任意一对顶点都不被 E中边相连。

独立集问题

- 定义（独立集）给定无向图 $G=(V,E)$，G 的独立子集为一个顶点子集 $V' \subseteq V$ 满足：

V'中任意一对顶点都不被 E中边相连。

- 独立子集 V 的规模为 V 中顶点个数。

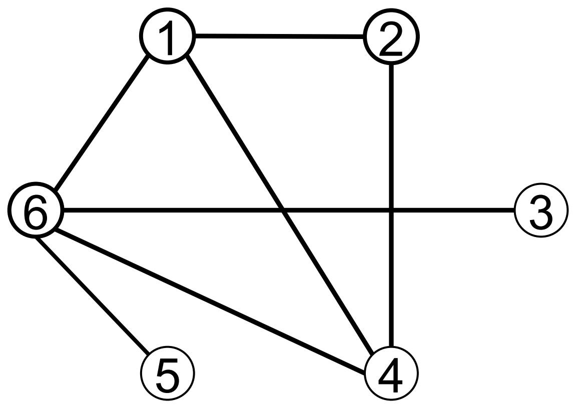
独立集问题

- 定义（独立集）给定无向图 $G=(V,E)$，G 的独立子集为一个顶点子集 $V'\subseteq V$ 满足：

  V'中任意一对顶点都不被 E中边相连。

- 独立子集 V 的规模为 V 中顶点个数。

例：



G

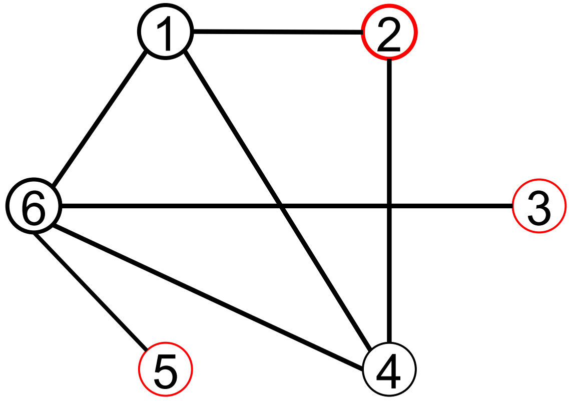
独立集问题

- 定义（独立集）给定无向图 $G=(V,E)$，G 的独立子集为一个顶点子集 $V' \subseteq V$ 满足：

  V'中任意一对顶点都不被 E中边相连。

- 独立子集 V 的规模为 V 中顶点个数。

例：



G

$\{2,3,5\}$ 是 G 的一个独立集

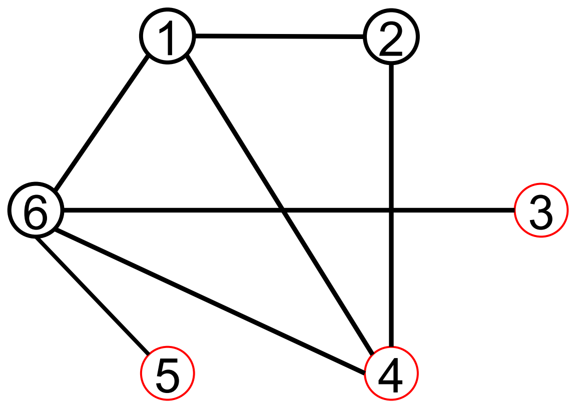
独立集问题

- 定义（独立集）给定无向图 $G=(V,E)$，G 的独立子集为一个顶点子集 $V'\subseteq V$ 满足：

V'中任意一对顶点都不被 E中边相连。

- 独立子集 V 的规模为 V 中顶点个数。

例：



G

$\{2,3,5\}$ 是 G 的一个独立集

$\{3,4,5\}$ 是 G 的一个独立集

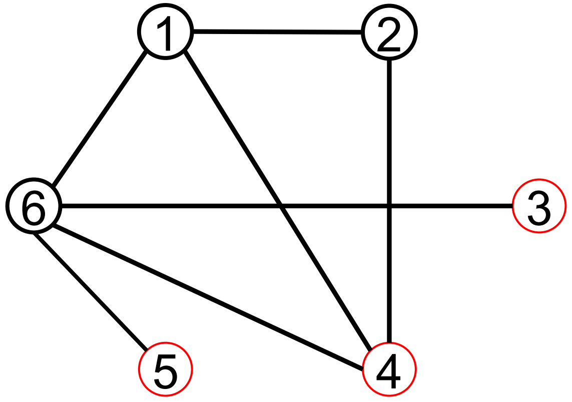
独立集问题

- 定义（独立集）给定无向图 $G = (V, E)$，G 的独立子集为一个顶点子集 $V' \subseteq V$ 满足：

  V'中任意一对顶点都不被 E中边相连。

- 独立子集 V 的规模为 V 中顶点个数。

例：



G

$\{2,3,5\}$ 是 G 的一个独立集

$\{3,4,5\}$ 是 G 的一个独立集

独立集的子集仍是独立集

独立集问题

- 独立集（DIS）问题
  - 输入：无向图 $G = (V, E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

独立集问题

- 独立集（DIS）问题
  - 输入：无向图 $G = (V, E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

- 定理：独立集问题是NP完全问题（DIS ∈ NPC）

证明思想：两步（上界与下界）

(1) 上界： DIS ∈ NP

(2) 下界：找到一个NP完全问题 L，使得 $L \le_P$ DIS

- DCLIQUE $\le_P$ DIS

上界证明：DIS ∈ NP

- 独立集（DIS）问题
  - 输入：无向图 $G=(V,E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

- DIS ∈ NP

证明： 对任意无向图 $G=(V,E)$和顶点子集 $V'\subseteq V$，其中 $|V'|=k$，下面证明可以在多项式时间内验证 V' 是一个独立集，

上界证明：DIS ∈ NP

- 独立集（DIS）问题
  - 输入：无向图 $G = (V, E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

- DIS ∈ NP

证明： 对任意无向图 $G = (V, E)$和顶点子集 $V' \subseteq V$，其中 $|V'| = k$，下面证明可以在多项式时间内验证 V' 是一个独立集，

即验证：对任意 $u, v \in V'$， $(u, v) \notin E$。

上界证明：DIS ∈ NP

- 独立集（DIS）问题
  - 输入：无向图 $G = (V, E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

- DIS ∈ NP

证明： 对任意无向图 $G = (V, E)$和顶点子集 $V' \subseteq V$，其中 $|V'| = k$，下面证明可以在多项式时间内验证 V' 是一个独立集，

即验证：对任意 $u, v \in V'$， $(u, v) \notin E$。

显然 ，当把G 表示为邻接矩阵时，以上验证可在 $O(|V|^2)$时间内完成。

上界证明：DIS ∈ NP

- 独立集（DIS）问题
  - 输入：无向图 $G=(V,E)$，整数 k
  - 问题： G 是否存在包含 k 个顶点的独立集

- DIS ∈ NP

证明： 对任意无向图 $G=(V,E)$和顶点子集 $V'\subseteq V$，其中 $|V'|=k$，下面证明可以在多项式时间内验证 V' 是一个独立集，

即验证：对任意 $u,v\in V'$， $(u,v)\notin E$。

显然 ，当把G 表示为邻接矩阵时，以上验证可在 $O(|V|2)$时间内完成。

因此，独立集问题是NP问题。

下界证明：DCLIQUE $\leq_P$ DIS

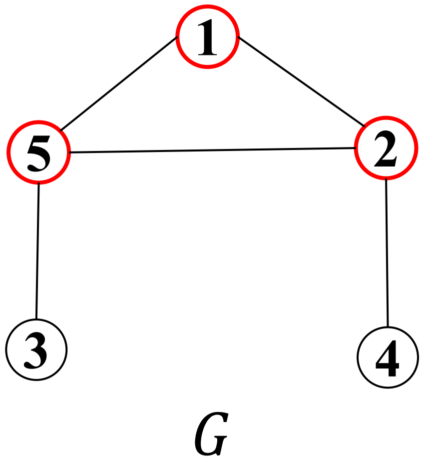
- 设 $G = (V, E)$是无向图，$V' \subseteq V$，则
  - V' 是G 的一个团 当且仅当 对任意 $v, v' \in V'$，$(v, v') \in E$；
  - V' 是G 的一个独立集 当且仅当 对任意 $u, v \in V'$， $(u, v) \notin E$。

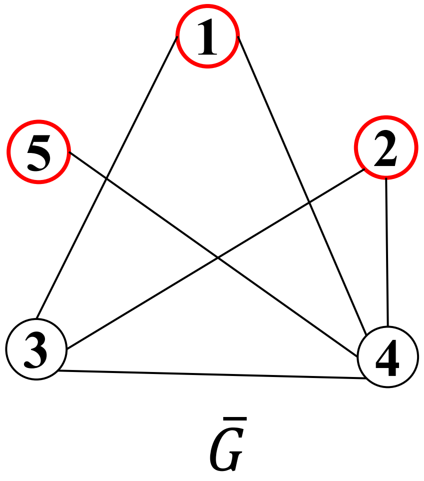
下界证明：DCLIQUE $\leq_P$ DIS

- 设 $G = (V, E)$是无向图，$V' \subseteq V$，则
  - V' 是G 的一个团 当且仅当 对任意 $v, v' \in V'$，$(v, v') \in E$；
  - V' 是G 的一个独立集 当且仅当 对任意 $u, v \in V'$， $(u, v) \notin E$。
- 构造 G 的补图 $\bar{G} = (V, \bar{E})$，其中 $(u, v) \in \bar{E}$ 当且仅当 $(u, v) \notin E$

下界证明：DCLIQUE $\leq_P$ DIS

- 设 $G=(V,E)$是无向图，$V' \subseteq V$，则
  - V' 是G 的一个团 当且仅当 对任意 $v,v' \in V'$，$(v,v') \in E$；
  - V' 是G 的一个独立集 当且仅当 对任意 $u,v \in V'$，$(u,v) \notin E$。
- 构造 G 的补图 $\bar{G}=(V,\bar{E})$，其中 $(u,v) \in \bar{E}$ 当且仅当 $(u,v) \notin E$

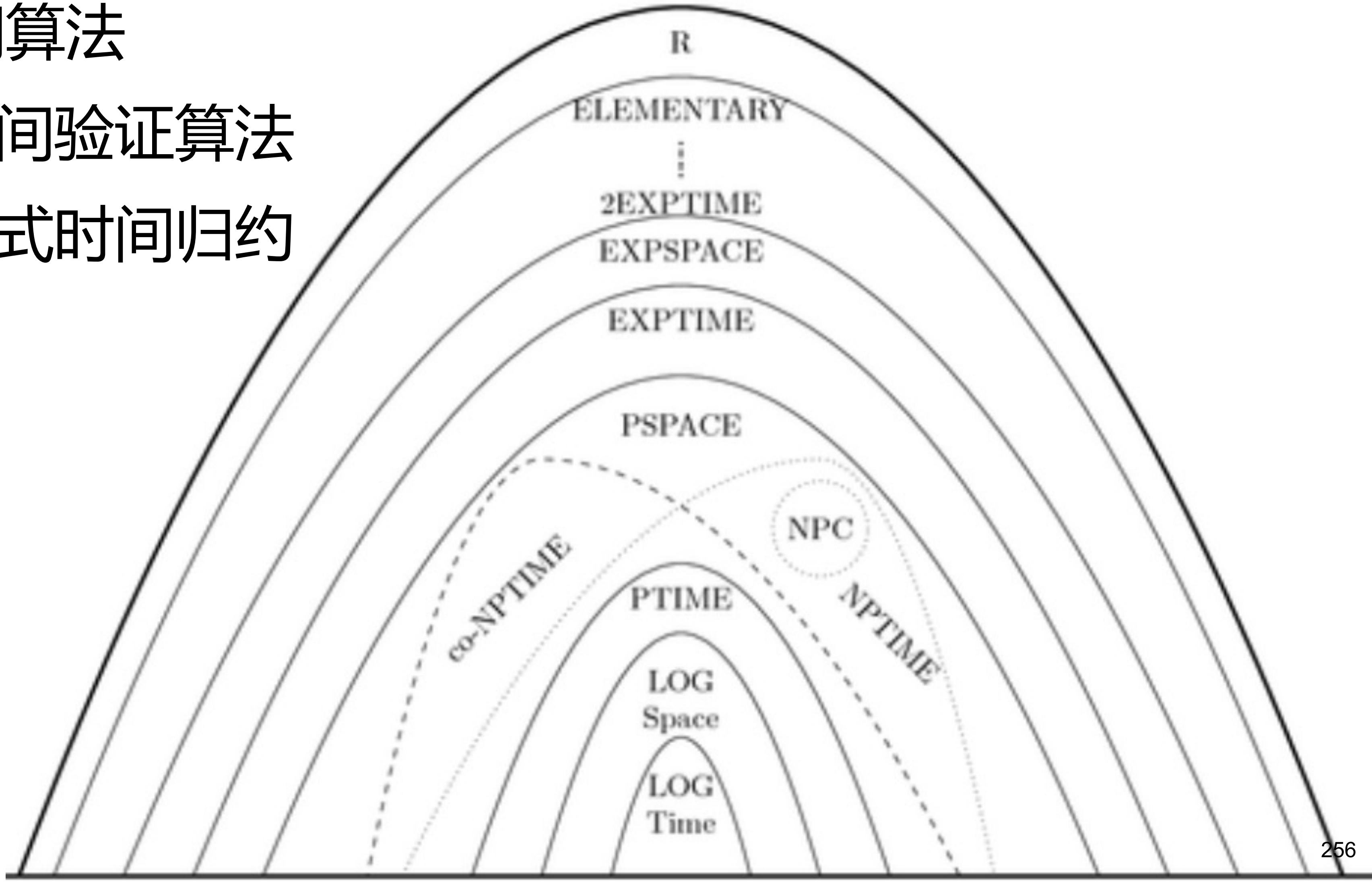


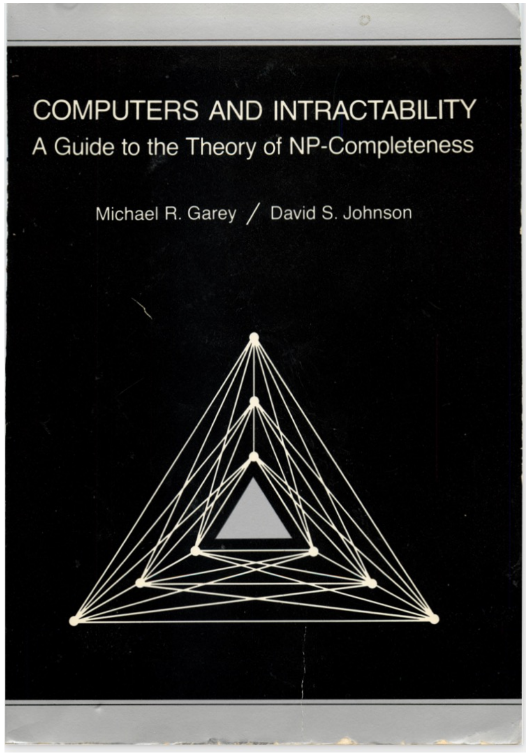


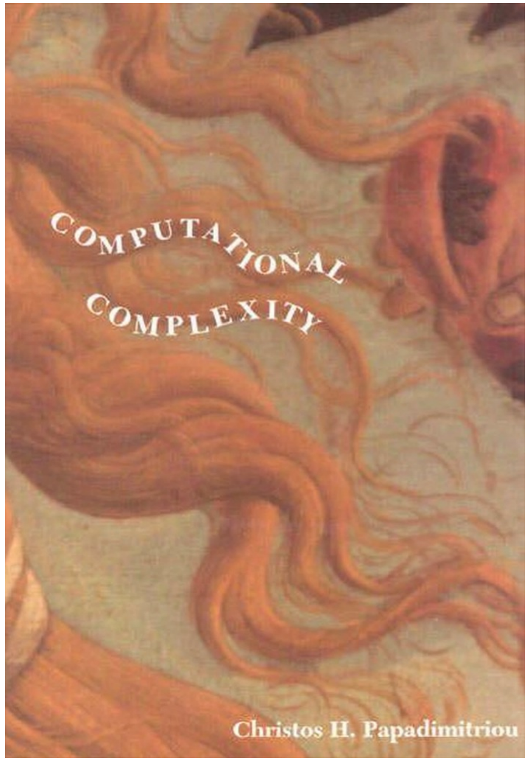
$\{1,2,5\}$ 是
- G 的一个团
- $\bar{G}$ 的一个独立集

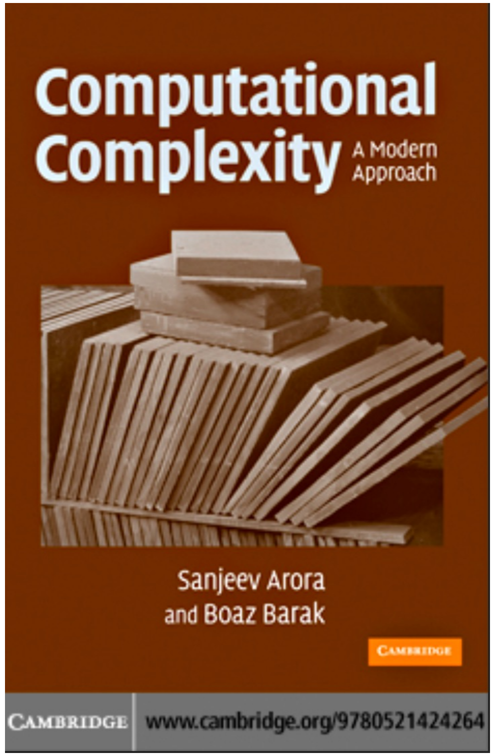
总结

- 判定问题
- P问题与多项式时间算法
- NP问题与多项式时间验证算法
- NP完全问题与多项式时间归约
- NP完全问题证明









- Michael R. Garey and David S. Johnson. Computers and intractability: a guide to the theory of NP-complete. 1979
- Christos H. Papadimitriou. Computational complexity. 1994
- Sanjeev Arora and Boaz Barak. Computational complexity: A modern approach. 2009