NP-Completeness

Jennya

2013/10/31

P

這些問題可以在polynomial time內解決

$O(n^k), for\ some\ k$

NP

這些問題可以在polynomial time內被verify

certificate

hamiltonian cycle problem

directed graph G = (V,E) + 〈 v1, v2, v3, … v|V| 〉

=> check if correct

NPC

NP-complete

$\theta(3^n), \theta(2^n)$

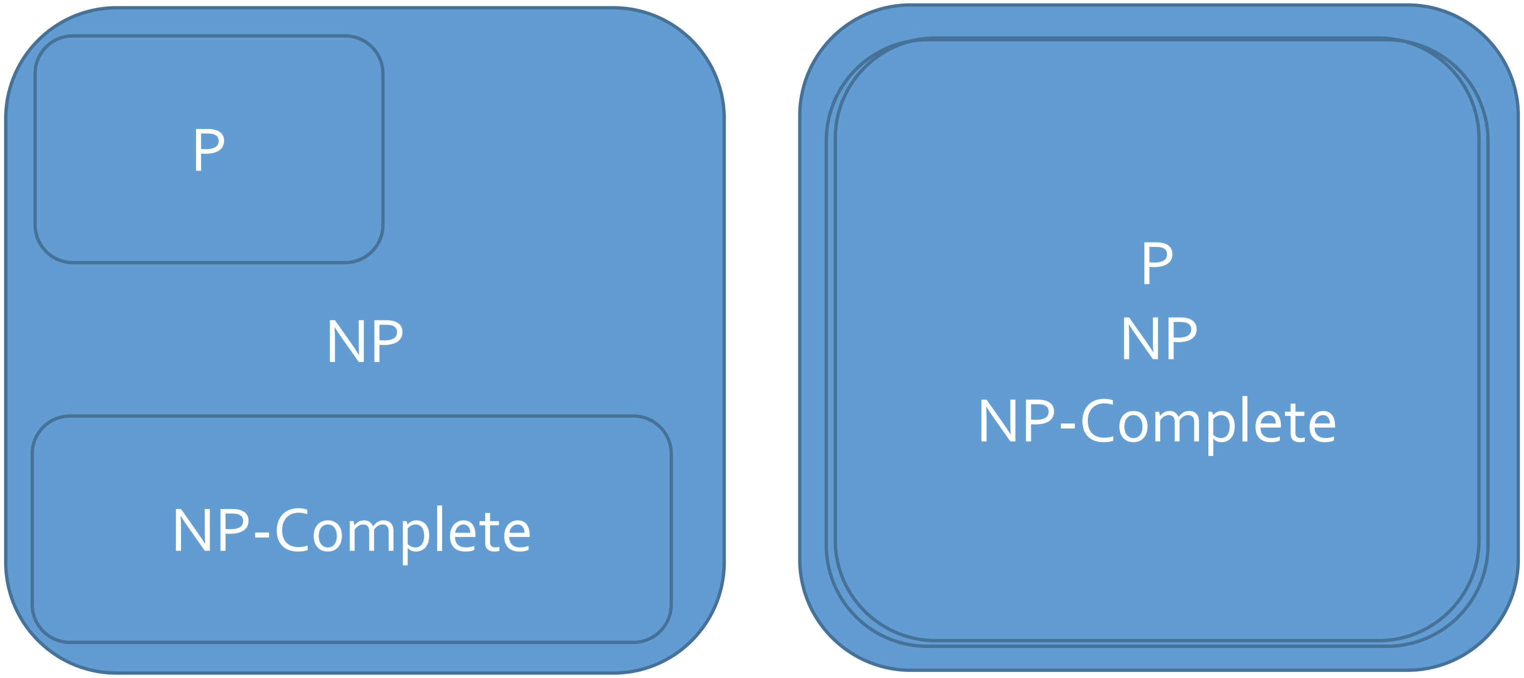
NP-Complete

- 一個問題是NP, 而且跟NP裡面的其他問題至少”一樣難”
- (一樣難: 所有NP裡面的問題可以在polynomial time裡面轉換成NP-complete的問題)
- 因為以上, 所以”一解則全解”

NP-Complete

1. 這些問題目前尚未找到可以在polynomial time內解決的algorithm.

2. 這些問題目前尚未被證明無法在polynomial time內解決.



P=NP???

學這個有什麼用?

- 雖然還沒有人證明出NP-complete problems無法在polynomial time裡面解出來
  但是經過了40年了, 沒有任何NP-complete problem被解出來(polynomial time)
- 因此證明某問題為NP-complete某種程度上也證明了它非常難解(或甚至無法解)
- 近似演算法 / 針對特別condition

key concepts in showing a problem to be NP-complete

- *Decision problems*
- *Reductions*
- *A first NP-complete problem*

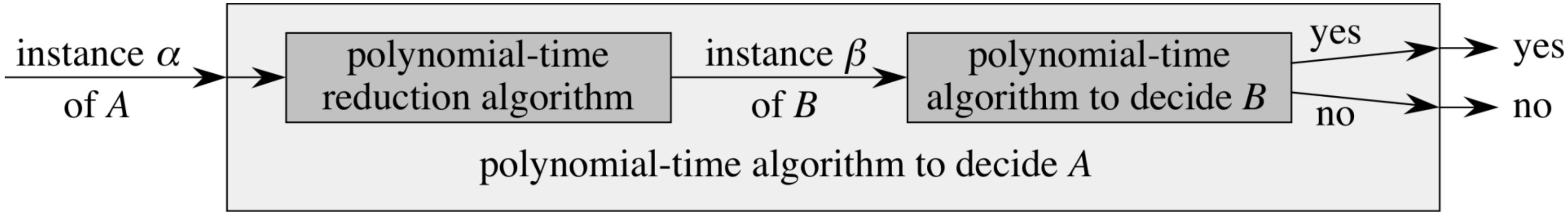
Decision problem v.s. optimization problem

- Decision problem:
  輸出是yes/no (1/0) (可不可以找到答案)
- Optimization problem:
  輸出是最好的解 (可以的答案中找出最好的那個)

Decision problem

- NP-Complete只適用於decision problems
- 怎麼將optimization problem轉換成decision problem呢?
- 對要optimize的值設定一個bound
- 將Shortest path轉換成decision problem:
- 例子: Shortest path v.s. Path
- 給定graph G, vertices u & v, integer k, 有沒有從u到v的路徑使用少於k個edge?

用Reduction證明”一樣難”



- 存在instances of A → instances of B
  的 polynomial-time reduction algorithm
- 假設A沒有polynomial-time algorithm exist
- 則B也沒有polynomial-time algorithm可以解

- 使用反證法,假設B有 → 則A也有

矛盾. 所以B也沒有.