NP完全性理论

内容

- 算法的难易
- P与 NP
- Np-hard与NPC

1	constant	`a = b + c;`	statement	add two numbers
log N	logarithmic	`while (N > 1) { N = N / 2; ... }`	divide in half	binary search
N	linear	`for (int i = 0; i < N; i++) { ... }`	loop	find the maximum
N log N	linearithmic	[see mergesort lecture]	divide and conquer	mergesort
N^2	quadratic	`for (int i = 0; i < N; i++) for (int j = 0; j < N; j++) { ... }`	double loop	check all pairs
N^3	cubic	`for (int i = 0; i < N; i++) for (int j = 0; j < N; j++) for (int k = 0; k < N; k++) { ... }`	triple loop	check all triples
2^N	exponential	[see combinatorial search lecture]	exhaustive search	check all subsets

$1,\ \log N,\ N,\ N \log N,\ N^2,\ N^3,$ and 2^N

suffices to describe order-of-growth of typical algorithms.

Typical orders of growth

算法的难易

- 了解一个问题的计算时间下界有助于确定算法的效率与改进余地
- 通常我们根据解决问题所需时间是问题规模的多项式函数或是指数函数来区分问题的“难”，“易”
  - 最常见的组合算法大致可分这两类
- 实际问题当中，人们对于很多问题无法了解其内在复杂性，故通过分类将计算复杂性大致相同的问题归类进行研究。而对能够彻底分析的尽可能准确的确定其计算复杂性

例1：货郎担问题
(Traveling salesman problem)

- 给定n个城市，任意两个城市间有路相连，一个货郎从一个城市出发，不重复的遍历所有的城市并回到起点，求一条路程最短的路径。
- 加权完全图$G=(V,E)$，$|V|=n, W:E\to R^+$ 求Hamilton圈C_h，使得 $W(C_h)=\sum_{e\in C_h} w(e)=\min\{W(C_i)\}$
- 计算复杂度：$O(n!)$

P类（Polynomial)

- 判定问题：只有肯定和否定两种答案
  - 优化问题可以化作判定问题处理
- P类
  - 具有多项式时间算法的判定问题形成的计算复杂性类
  - 猜测TSP（Traveling salesman problem)不属于P（J.Edmonds 1965）

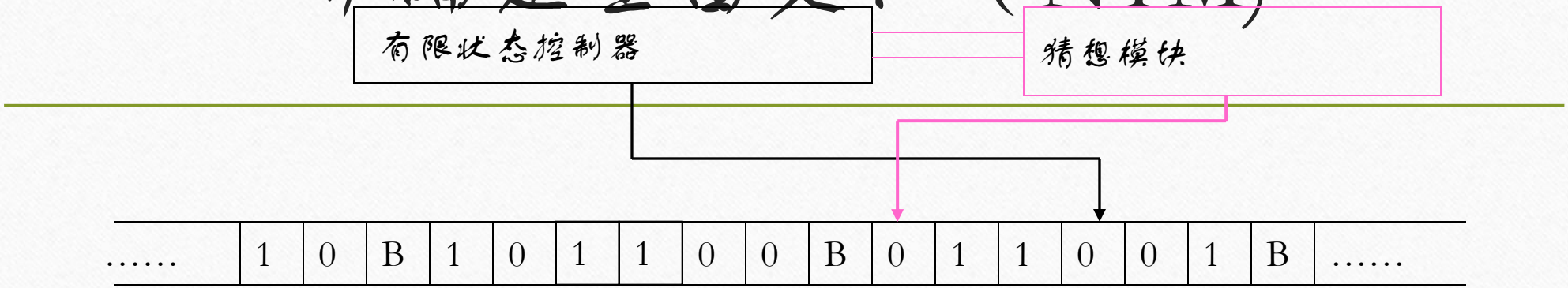
非确定性计算模型

- 类似上述两个例子人们一直没能找出确定性的算法可以在多项式时间范围内解决该问题
- 也不能证明这样的问题一定需要超多项式时间下界
- 人们提出了非确定性图灵机计算模型，使得许多问题可以在多项式时间内求解

非确定型算法

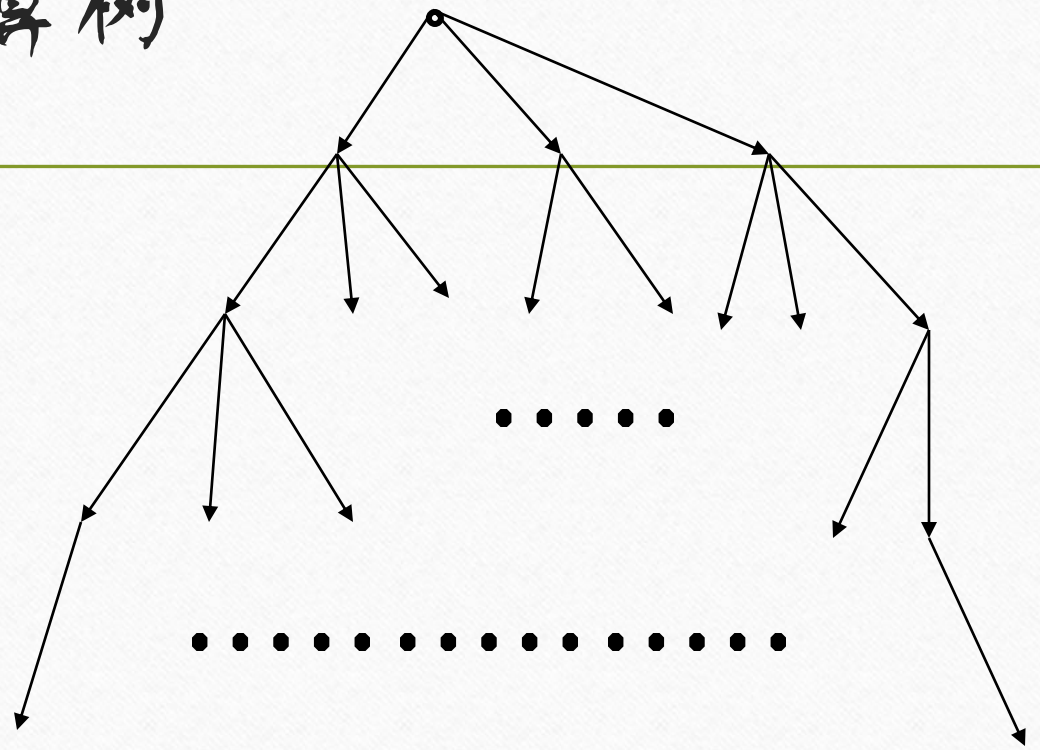
- 不现实的计算
  - 现实中的计算方式都是确定的
- 解TSP问题的一个非确定型算法
  - 第一步：猜测一个赋值；
  - 第二步：检查该赋值是否满足
- 非确定型算法的计算时间：
  - 各种可能的计算过程的最短时间

非确定型图灵机（NTM）



- 猜想阶段
- 验证阶段

NTM计算树



计算过程：从根到叶节点的路径

NP类 (Nondeterministic Polynomial)

- NP问题：
  - 在非确定型图灵机上多项式时间可解的问题
  - 在确定型图灵机上多项式时间可验证的问题
- P类包含于NP类中
- NP类问题在确定图灵机上指数时间可解
  - 非确定型图灵机和确定型图灵机的计算能力相当

计算难度比较的标准

- 难易是比较而言的
  - 多项式时间归约（Karp归约 1972)
- 定义
  - 问题A多项式时间内转化为问题B的特殊情况，则称A可多项式归约于B，记为 $A \leq_p B$
    - 转化时间为多项式
    - 对于A的输入*I*的回答与其对应的B的输入 *f(I)* 一致

NP完全与NP-hard

- NP完全问题：$p \in NPC : 1.p \in NP;\ 2.\forall q \in NP,$有$q \leq_p p$
- NP-hard问题：$p \in NP - hard :\ \forall q \in NP,$有$q \leq_p p$

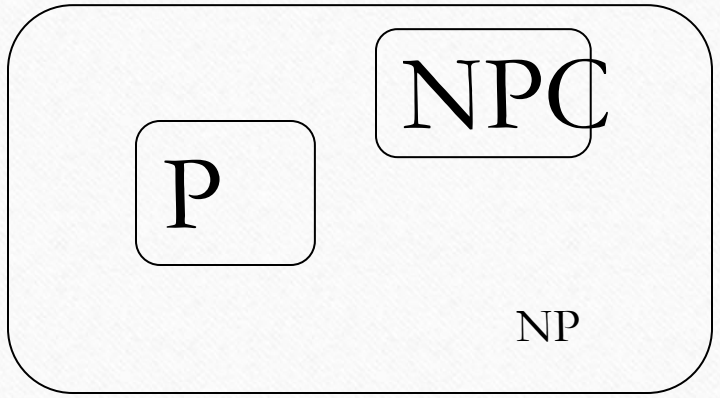
NP完全问题

- 第一个NP完全问题（Cook定理 1971）
  - 可满足性问题是NP完全问题
- 六个NP完全问题（Karp 1972)
  - 3SAT，3DM，VC，团，HC，划分
- 更多的NP完全问题
  - 1979年：300多个
  - 1998年：2000多个

P=?NP （P-NP问题）

$$P \bigcap NPC \neq \phi \Rightarrow P = NP$$

P=NP



现在的估计

如果 $P \neq NP$ ，则指数灾难无法避免

如何处理NPC问题

- 并行计算
  - 以硬件换取时间（理想模型可以多项式时间求解问题）
  - 实际困难（处理器数目，通讯代价等）
- 随机算法
  - 判定问题：
    - 以较大概率得到正确输出
    - 输出正确，但计算时间不定
  - 优化问题：输出解的性能不稳定
    - 以较大概率得到性能好的解

如何处理NPC的方法

- 完全算法
  - 规模较小可以直接计算
  - 寻找部分能在多项式时间求解的特殊情况
- 近似算法
  - 追求较优

參考閱讀

- http://episte.math.ntu.edu.tw/articles/mm/mm_10_2_04/index.html
- https://zh.wikipedia.org/wiki/P/NP%E9%97%AE%E9%A2%98