

计算复杂性

"基于有穷观点的能行方法"的"可计算"概念

仅仅涉及到问题的解决**是否能在有限资源内**(运行时间/运行空间)完成

并不关心具体花费多少计算步骤或多少*存储*空间

由于资源（时间/空间）相当有限，对于问题的解决需要考虑其可行性如何

人们发现各种不同的**可计算**问题，其难易程度是不一样的

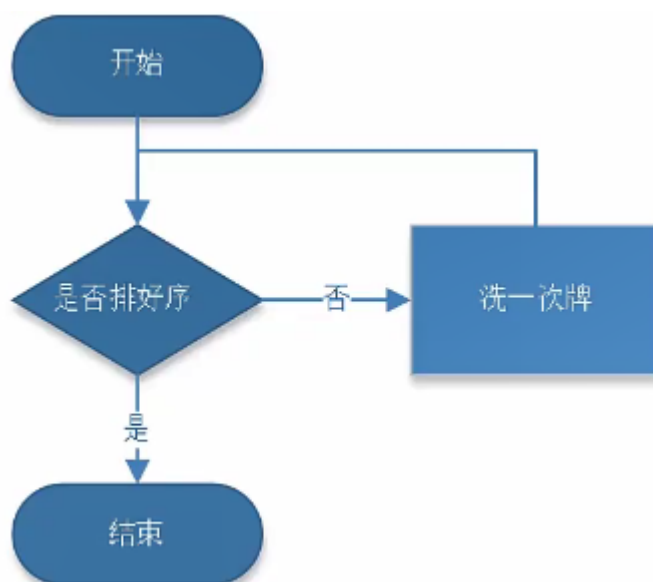
- 有些问题**非常容易**解决，如基本数值计算；
- 有些问题的解决程度**尚能令人满意**，如表达式求值、排序等；
- 有些问题的解决会爆炸性地吞噬资源，虽有解法但**没什么可行性**，如哈密顿回路、货郎担问题等

定义一些衡量指标,对问题的难易程度(所需的执行步骤数/存储空间大小)进行 **分类**,是计算复杂性理论的研究范围

但对于同一个问题，也会有不同的解决方案，其解决效率上也是千差万别如排序问题，以n张扑克牌作为排序对象般人们会想到的是“冒泡”排序，即每次从牌堆里选出一张最小的牌，这样全部排完大概会需要 n^2 量级的比较次数

另一种有趣的"Boog"排序方法

洗一次牌,看是否排好序,没有的话,接着洗牌,直到排序成功 全部拍完平均需要 $n*n!$ 量级的比较次数



(流程图)

计算复杂性与算法

计算复杂性理论研究问题的本质,将各种问题按其难易程度**分类**,研究各类问题的难度**级别**,并不关心解决问题的具体方案

算法则研究问题再**不同现实资源约束**情况下的不同解决方案,致力于找到**效率**最高的方案

不同硬件配置

不同运行环境

不同应用领域

甚至不同使用状况

不可计算问题

有不少定义清晰,但无法解决的问题

并不是尚未找到解,而是在"**基于有穷观点的能行方法**"的条件下,已被证明并**不存在**解决方案

停机问题:判定任何一个程序在任何一个输入情况下是否能够**停机**(不可计算问题)

不可计算数:

几乎所有的**无理数**,都无法通过算法来确定其任何一位是什么数字

可计算数很少:如圆周率 π ,自然对数的底 e