Design and Analysis of Algorithms
Part VII: Review
Lecture 12: Review

Yongxin Tong (童咏昕)

School of CSE, Beihang University

yxtong@buaa.edu.cn

期末考试概述

- 时间与地点
  - 2017年12月26日(周二晚)， 2小时
  - 具体地点待定
- 考试题型
  - 选择题+判断题(20% - 25%)
  - 算法运行实例+算法设计题(75% - 80%)
- 考试范围
  - 不涉及二分图匹配部分
  - 不涉及NPC的证明

Outline

- 基础知识
- 分治算法(Divide and Conquer)
- 图算法(Graph Algorithms)
- 动态规划(Dynamic Programming)
- 贪心算法(Greedy Algorithms)
- 处理难问题(P, NP and NPC)
- 例题

基础知识

- 渐进符号： O, Ω, Θ
- 最差情况分析，期望情况分析，平均情况分析
- 和式：$\sum_{i=1}^{n} i, \sum_{i=1}^{n} i^2, \sum_{i=1}^{n} \frac{1}{i}, \sum_{i=1}^{n} c_i \; (c < 1, c = 1, c > 1)$
- 递归式
  - 代入法(Substitution method)
  - 猜测并归纳(Guess and induction)
  - Example: $\mathrm{T(n) = 4T\left(\frac{n}{2}\right) + n, if\ n > 1}$

分治算法

- 一些常用的递归式结果
  - $O(\log n): T(n) = T\left(\frac{n}{2}\right) + O(1)$
  - $O(n): T(n) = T\left(\frac{n}{2}\right) + O(n)$
  - $O(n \log n): T(n) = 2T\left(\frac{n}{2}\right) + O(n)$
  - $O(n^c): T(n) = a \cdot T\left(\frac{n}{b}\right) + O(n)$
- 分治与随机算法结合

图算法

- 广度优先搜索及应用(BFS)
  - 连通分量(Connected Components)
  - 无权重图上的最短路径(Shortest Path in Unweighted Graphs)
- 深度优先搜索及应用(DFS)
  - 连通分量(Connected Components)
  - 拓扑排序(Topological Sorting)
- 最小生成树(Minimum Spanning Trees)
  - Prim算法及优先队列(Priority Queue)
  - Kruskal算法及并查集(Union-Find Set)
- Dijktra算法(Dijkstra’s Shortest Path Algorithm)
  - 与Prim算法的联系与区别
  - 不支持负权重边的图

动态规划

- 问题分析过程
  - 分析子问题空间(Space of Subproblems)
  - 建立递归式(Recurrence)
  - 自底向上计算(Bottom-Up Computation)
  - 记录最优方案的详细信息(Construction of the Optimal Solution)

贪心算法

- 算法通常较简单
- 难点在于如何证明算法的正确性
  - 一个常见的方法是：假设最优方案为X，而贪心算法得到的结果为Y。若X与Y不同，则能够在保证X不变差的情况下将X转化为Y。

处理难问题

- 输入规模：使用自然编码，不需考虑多项式上的差别，而只需关心$O(\log n)$及poly(n).
- 问题分类：P, NP and NPC
  - 定义
  - 已知$P \subseteq NP, NPC \subseteq NP$, 但是“=”是否成立仍未知。
- 证明问题Y为NP-Completeness
  - 证明$Y \in NP$
  - 寻找一个已知的NP-Complete问题X, 并证明$X \leq_p Y$.
  - 如何证明$X \leq_p Y$? 使用Y的结果解决X
    - 对问题X的任意一个输入x, 将其映射为Y的一个输入f(x)。
    - 证明问题X在输入x的条件下返回”yes”当且仅当问题Y在输入f(x)的条件下返回”yes”.


dank u
ju faleminderit
Tack
Asante
谢谢
Tak
mulţumesc
kiitos
Salamat!
Gracias
Terima kasih
Aliquam
Merci
Dankie
Obrigado
ありがとう
köszönöm
grazie
Aliquam
Go raibh maith agat
děkuji
Thank you
gam