

1 复习

1.1 题型

- 简答 8×8
- 综合 3×12

注意作业, 算法思路怎么写?

1.2 大纲

1.2.1 概述 (第一章)

算法 解决问题的一种方法或一个过程, 包括输入输出, 确定性和有限性

程序 算法用某种程序设计语言的具体实现, 且不必有限

时间复杂度, 渐进符号 $\omega < \Omega \leq \Theta \leq O < o$

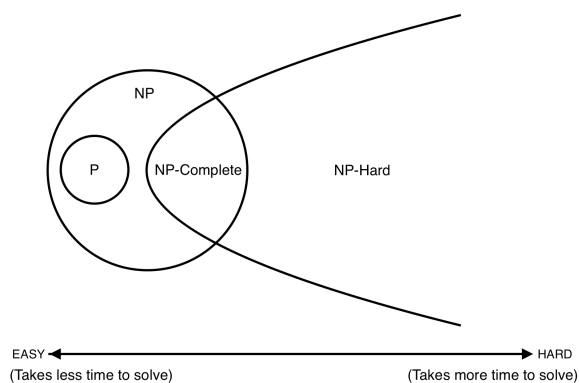
1.2.1.1 P/NP

P 多项式时间内可解决的判定问题

NP 在多项式时间内可验证的判定问题

[NPC] NPC 是一个 NP 问题, 任意一个 NP 问题都可以多项式时间内可归约到 [NPC]

[NPH] 任意一个 NP 问题都可以多项式时间内可归约到 [NPH]



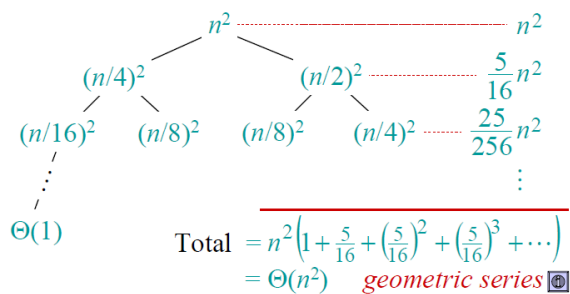
1.2.1.2 时间复杂度

常见时间复杂度 $O(1)$ (奇偶判断) $< O(\log n)$ (二分查找) $< O(n^{\frac{1}{2}})$ $< O(n)$ $< O(n * \log n)$ $< O(n^2)$ $< O(n^3) \leq P$ (多项式时间) $< O(2^n)$ $< O(n!)$

$$T = kT(n/m) + f(n) = O(n^{\log_m k}) + O(f(n))$$

Example of recursion tree

Solve $T(n) = T(n/4) + T(n/2) + n^2$:



1.2.2 递归与分治 (第二章)

1.2.2.1 二分搜索

时间复杂度 while 循环被执行了 $O(\log n)$ 次, 循环体内需要 $O(1)$ 时间, 因此为 $O(\log n)$

1. 分个数相近的 2 部分 [$<x$], x , [$>x$]]

1.2.2.2 棋盘覆盖

1. 用 3L 骨牌, 分 4 个特殊骨牌

1.2.2.3 循环赛日程表

1. 分配 $n/2$ 个选手, 直到两个选手

1.2.3 DP (第三章)

例 1.2.1 DP 与分治区别?

解

- 都是由下向上, 过程都是递归的

- 但动态规划避免了重复计算

1.2.3.1 矩阵连乘

$$O(n^3)$$

$$\min[i:j] = \begin{cases} 0 & i = j \\ \min\{\min[i:k] + \min[k+1:j] + p_{i-1}p_kp_j\} & i < j \end{cases}$$

1.2.3.2 流水作业调度

1. $N_1 = \{i \mid a_i < b_i\}, N_2 = \{i \mid a_i \geq b_i\}$
2. N_1 按 a_i 非减序, N_2 按 b_i 非增序
3. 一个个来

1.2.3.3 01 背包问题

```
m :: [(Int, Int)] -> Int -> Int
m ((w,v):[]) size | w > size = 0
                    | otherwise = v
m ((w,v):xs) size | w > size = m xs size
                    | otherwise = max (m xs size) ((m xs $ size - w) + v)
```

1.2.4 贪心算法 (第四章)

什么是贪心算法? 能找到贪心条件, 但不一定是最优解

1.2.4.1 哈夫曼编码

每次选择频率最高的字符构造树

1.2.4.2 单源最短路径

顶点集合 S , 不断贪心选择到集合外元素路径中最短的来扩充这个集合, 当 S 中包含了所有顶点即可

贪心选择性质 算法贪心选择具有最短特殊路径顶点 u (即 $\text{dist}[u]$ 是源外 dist 集合中最小的元素). 若存在一点 x 在源集合外, 这条路从 S 出发到 x , 再在 S 内外徘徊若

干次, 最后离开 S 到 u . 证明不存在能使这条路径能比 $dist[u]$ 更近.

假设存在这么一条路径. 即 $d(v, u) = d(v, x) + d(x, u) < dist[u]$

易得 $dist[x] \leq d(v, x)$

由于 $d(v, u) > 0$, 可得 $dist[x] < dist[u]$

与条件矛盾, 得证

最优子结构性质 证明 $dist[u]$ 即当前源到 u 的最短路径.

对于源外的点 u :

若此时贪心选择了 i (即 $dist[i]$ 为当前源外 $dist$ 中最小值), 存在一条 i 到 u 的路径, 且 $dist[i] + c[i][u] < dist[u]$

那么 $dist[i] + c[i][u]$ 将作为 $dist[u]$ 的新值, 显然成立.

对于源内的点 u :

若此时贪心选择了 i (即 $dist[i]$ 为当前源外 $dist$ 中最小值), 存在一条 i 到 u 的路径, 且 $dist[i] + c[i][u] < dist[u]$

那么 $dist[i] < dist[u]$, 与条件矛盾. 即不存在 $dist[i] + c[i][u] < dist[u]$.

综上得证

1.2.4.3 最小生成树

1. Prim: 从树内顶点到树外顶点的边中选最短的一个
2. Kruskal: 从连接不同连通分支的边中选最短的一个

1.2.4.4 多机调度问题

优先大作业

1.2.5 回溯法, 分支限界法 (第五, 六章)

{ DFS, BFS } + 减支 + 动态生成解空间树

1.2.5.1 回溯法

01 背包问题 DFS 完全二叉树. 到达某节点时若超重则剪去

旅行售货员问题 DFS 地图. 到某节点时若耗时比已知最短距离长则剪去

1.2.5.2 分支限界法

单源最短路径 BFS 图. 可选用优先队列 BFS

到某节点时, 若比已知到此节点的距离长, 或已知最短到达目标的距离长, 则剪去

1.2.6 随机化算法

1.2.6.1 PI 值

```
4 * definite_integral(lambda x: sqrt(1 - x**2))
```

1.2.6.2 定积分

```
def definite_integral(f, n = 100000):  
    k = 0  
    for i in range(n):  
        x, y = random(), random()  
        if f(x) >= y:  
            k += 1  
    return k / n
```