Ch0 介绍

1. 基本信息

1.1 教学用书

Algorithm Design by Jon Kleinberg and Éva Tardos

1.2 课程评价

- 期末考试 40%
- Lab 30%
- 作业 20%
- 考勤 10%

2. 教学大纲

- 贪心 **Greedy**
- 分治 **Divide-and-conquer**
- 动态规划 **Dynamic programming**
- 网络流 **Network flow**
- 随机算法 Randomized algorithms
- NP-Hard 问题解决方法 Coping with intractability

3. 归纳法 Induction

3.1 数学归纳法

数学归纳法是一种证明命题 $P(n)$ 对所有自然数 n，或自然数的某个无穷子集（例如，所有正的偶数）都成立的技术

3.2 证明步骤

- Claim：$P(n)$ 对于所有的正整数 n 都是正确的
- Proof：我们对 n 进行归纳
- Base：我们需要证明 $P(1)$ 是正确的
- Induction：假设 $P(n)$ 对于 $n = 1, 2, \ldots, k-1$ 都是正确的，我们需要证明 $P(k)$ 是正确的

3.3 例题

- Claim：证明对于任何正整数 n，$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- Proof：我们需要证明 $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ 适用于任何正整数 n，使用归纳法
- Base：当 $i = 1$ 时，$\sum_{i=1}^{1} i = \frac{1\times2}{2} = 1$，满足
- Induction：
  - 假设 $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ 对于任何 $n = 1, 2, \ldots, k-1$ 都成立
  - 我们需要证明的是 $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$
  - 我们知道 $\sum_{i=1}^{k-1} i = \frac{(k-1)k}{2}$ 是成立的，且 $\sum_{i=1}^{k} i = \sum_{i=1}^{k-1} i + k$
  - 即 $\frac{(k-1)k}{2} + k = \frac{k(k+1)}{2}$
  - 故得证

4. 递归法 Recursive

数学中的递归函数定义与编程语言中的递归过程基本相似

递归定义总是包含两部分

- 基本情况
- 递归公式

基本情况和递归公式都必须存在才能有完整的定义

当 n 的结果依赖于大于一个较小值的结果时，递归定义的真正威力就显现出来了，例如下面的示例

4.1 例题 1

$\sum_{i=1}^{n} i$ 可以被定义成

- $g(1) = 1$

- $g(n) = g(n-1) + n$ （对于所有 $n \geq 2$）

4.2 例题 2

斐波那契数列

- $F_0 = 0$
- $F_1 = 1$
- $F_i = F_{i-1} + F_{i-2} \quad \forall i \geq 2$

如何用非递归的方式表达这个模式并不明显

4.3 找到闭集 closed form

归纳法有的展开方法，来展开递归表达式的真实数学解，有两种找到闭集的方法

代入计算

找到下列递归表达式的闭集

- $T(1) = 1$
- $T(n) = 2T(n-1) + 3 \ \forall n \geq 2$

可以通过代入嵌套计算来完成

$$
\begin{aligned}
T(n) &= 2T(n-1) + 3 \\
&= 2(2T(n-2) + 3) + 3 \\
&= 2(2(2T(n-3) + 3) + 3) + 3 \\
&= 2^3 T(n-3) + 2^2 \cdot 3 + 2 \cdot 3 + 3 \\
&= 2^4 T(n-4) + 2^3 \cdot 3 + 2^2 \cdot 3 + 2 \cdot 3 + 3 \\
&\dots \\
&= 2^k T(n-k) + 2^{k-1} \cdot 3 + \dots + 2^2 \cdot 3 + 2 \cdot 3 + 3 \\
&= 2^k T(n-k) + 3\left(2^{k-1} + \dots + 2^2 + 2 + 1\right) \\
&= 2^k T(n-k) + 3 \sum_{i=0}^{k-1} \left(2^i\right)
\end{aligned}
$$

当 $n-k=1$ 即 $k=n-1$ 时候，我们到达 base case

$$
\begin{aligned}
T(n) &= 2^k T(n-k) + 3\sum_{i=0}^{k-1}\left(2^i\right) \\
&= 2^{n-1}T(1) + 3\sum_{i=0}^{n-2}\left(2^i\right) \\
&= 2^{n-1} + 3\sum_{k=0}^{n-2}\left(2^k\right) \\
&= 2^{n-1} + 3\left(2^{n-1}-1\right) = 4\left(2^{n-1}\right) - 3 = 2^{n+1} - 3
\end{aligned}
$$

假设法

找到下列递归表达式的闭集

- $f(0) = 2$
- $f(1) = 3$
- $f(n+1) = 3f(n) - 2f(n-1)\ \ \forall n \geq 1$

我们假设，$\forall n \in N, f(n) = 2^n + 1$

- $f(0) = 2^0 + 1 = 2$
- $f(1) = 2^1 + 13$
- $f(n+1) = 3 \times (2^n + 1) - 2 \times (2^{n-1} + 1) = 2^{n+1} + 1$

故得证

5. 反证法 Contradiction

反证法是一种强大的证明技术，在适当的情况下非常有用

然而，反证法的证明往往比直接的证明或矛盾的证明更缺乏说服力，也更难写

5.1 证明步骤

在反证法中，我们通过证明一个命题的否命题 $\neg P$ 是假的来反正一个命题 P 是正确的

如果 $\neg P$ 导致矛盾，那么 $\neg P$ 不可能为真，因此 $\neg P$ 一定为真

5.2 例题

Claim：没有最大的偶数

Proof：

- 假设，也就是说，假设有一个最大的偶数，我们称它为 k
- 因为 k 是偶数，所以它的形式是 $2n$，其中 n 是整数
- 考虑 $k+2$，$k+2=(2n)+2=2(n+1)$
- 所以 $k+2$ 是偶数，但是 $k+2$ 比 k 大
- 这与我们假设 k 是最大的偶数相矛盾，所以我们最初的说法一定是正确的