递推方程的求解

基本方法

- 迭代法
  - 直接迭代
  - 换元迭代
  - 差消迭代
  - 迭代树：代价 = sum(每层的代价，层数（一般是 $(logn)$））
- 尝试法
  尝试不同量级的解，分别代入迭代方程左右两边。可以尝试的解的形式有：
  - $T(n) = c$
  - $T(n) = cn$
  - $T(n) = cn^2$
  - $T(n) = cnlogn$
- 主定理

主定理

设 $a \geq 1$，$b \geq 1$ 为常数，$f(n)$ 为函数，$T(n)$ 为非负整数，且

$$T(n) = aT(\frac{n}{b}) + f(n)$$

则有：

1. 若 $f(n) = O(n^{log_b a - \epsilon}), \epsilon > 0$，那么 $T(n) = \Theta(n^{log_b a})$
2. 若 $f(n) = \Theta(n^{log_b a})$，那么 $T(n) = \Theta(n^{log_b a} logn)$
3. 若 $f(n) = \Omega(n^{log_b a + \epsilon}), \epsilon > 0$，且对于某个常数 $c < 1$ 和充分大的 n 有 $af(n/b) \leq cf(n)$ 那么 $T(n) = \Theta(f(n))$

特殊函数

- 取整函数
  - 求解方法：先猜想解的形式（直接去掉取整符号化成普通函数），然后用数学归纳法证明。
  - 例如：

    $$T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$$

    $$T(1) = 1$$

    首先根据

    $$T(n) = 2T(\frac{n}{2}) + n$$

    的解为

    $$T(n) = O(nlogn)$$

    因此可以首先猜想原递推方程的解的阶是 $O(nlogn)$，然后用数学归纳法进一步证明。