迭代法求解
递推方程

迭代法

- 不断用递推方程的右部替换左部
- 每次替换，随着 n 的降低在和式中多出一项
- 直到出现初值停止迭代
- 将初值代入并对和式求和
- 可用数学归纳法验证解的正确性

Hanoi 塔算法

$$T(n) = 2\,T(n-1) + 1$$
$$T(1) = 1$$

$$
\begin{aligned}
T(n) &= 2\,\underline{T(n-1)} + 1 \\
&= 2\,[\,\underline{2T(n-2) + 1}\,] + 1 \\
&= 2^2\,T(n-2) + 2 + 1 \\
&= \ldots \\
&= 2^{n-1}T(1) + 2^{n-2}+2^{n-3}+\ldots+2+1 \\
&= 2^{n-1} + 2^{n-1}-1 \\
&= 2^n-1
\end{aligned}
$$

代入初值
求和

插入排序算法

$$\begin{cases} W(n) = W(n-1) + n - 1 \\ W(1) = 0 \end{cases}$$

$$\begin{aligned}
W(n) &= W(n-1) + n-1 \\
&= [W(n-2) + n-2] + n-1 \\
&= W(n-2) + n-2 + n-1 \\
&= \ldots \\
&= W(1) + 1 + 2 + \ldots + (n-2) + (n-1) \\
&= 1 + 2 + \ldots + (n-2) + (n-1) \\
&= n(n-1)/2
\end{aligned}$$

换元迭代

- 将对 n 的递推式换成对其他变元 k 的递推式
- 对 k 直接迭代
- 将解 (关于 k 的函数) 转换成关于 n 的函数

二分归并排序

MergeSort (A, p, r)

输入：数组 $A[p..r]$

输出：按递增顺序排序的数组 A

1. **if** $p < r$
2. **then** $q \leftarrow \lfloor (p+r)/2 \rfloor$
3. **MergeSort** (A, p, q)
4. **MergeSort** $(A, q+1, r)$
5. **Merge** (A, p, q, r)

换元

假设 $n=2^k$，递推方程如下：

$$W(n)=2W(n/2)+n-1$$

$$W(1)=0$$

换元：

$$W(2^k) = 2W(2^{k-1}) + 2^k-1$$

$$W(0) = 0$$

迭代求解

$$\boxed{W(2^k) = 2W(2^{k-1}) + 2^k - 1}$$

换元，
对k迭代

$$
\begin{aligned}
W(n) &= 2\underline{W(2^{k-1})} + 2^k - 1 \\
&= 2[\underline{2W(2^{k-2}) + 2^{k-1} - 1}] + 2^k - 1 \\
&= 2^2\underline{W(2^{k-2})} + 2^k - 2 + 2^k - 1 \\
&= 2^2[\underline{2W(2^{k-3}) + 2^{k-2} - 1}] + 2^k - 2 + 2^k - 1 \\
&= \ldots \\
&= 2^k W(1) + k2^k - \boxed{(2^{k-1} + 2^{k-2} + \ldots + 2 + 1)} \\
&= k2^k - 2^k + 1 \\
&= n\log n - n + 1
\end{aligned}
$$

解的正确性–归纳验证

证明:下述递推方程的解是 $W(n)=n(n-1)/2$

$$W(n)=W(n-1)+n-1$$
$$W(1)=0$$

方法：数学归纳法

证　$n=1$，$W(1)=1\times(1-1)/2 = 0$

假设对于n，解满足方程，则

$$\begin{aligned} &W(n+1) \\ &= \underline{W(n)}+n = \underline{n(n-1)/2} + n \\ &= n[(n-1)/2+1] = n(n+1)/2 \end{aligned}$$

小结

迭代法求解递推方程

- 直接迭代，代入初值，然后求和
- 对递推方程和初值进行换元，然后求和，求和后进行相反换元，得到原始递推方程的解
- 验证方法——数学归纳法