

Bidirectional Increasing-Decreasing Subsequence

双向递增递减子序列

定义:

递减子序列和递增子序列的概念相同,但渐变方向相反,递减子序列的元素之间依次递减。

问题:

在长度为 n 的序列 s (范围是 $[1, n]$) 中寻找元素 $s[i]$, 使得 $s[1, i]$ 中的最长递增子序列和 $s[i, n]$ 中的最长递减子序列, 它们的长度总和最大。

解法:

序列 s 的长度为 n (数组从1开始, 范围为 $[1, n]$)。设 $f(i)$ 是以 $s[i]$ 作为最右边元素的最长递增子序列的长度, $g(i)$ 是以 $s[i]$ 作为最左边元素的最长递减子序列的长度。

最后返回 $\max\{f(i) + g(i) - 1\}$ (其中 $i \in [1, n]$), 即所有 $f(i) + g(i) - 1$ 中的最大值, 之所以减去1 是因为 $s[1, i]$ 最右边的元素和 $s[i, n]$ 最左边的元素是同一个元素, 重复了因此长度减1。该算法的时间复杂度是 $O(n^2)$ 。