

Q01: 什么是最长公共子序列 (LCS) ?

- prompt: 请详细定义“最长公共子序列” (LCS) 的含义, 并通过一个实例说明其应用。
- model_name: ChatGPT
- model_version: GPT-4
- response: LCS (Longest Common Subsequence) 是指两个序列的最长子序列, 且该子序列需要保持在两个序列中的相对顺序。注意, 它不要求连续。例如, 若有序列 $X=abcdefg$ 和 $Y=aaadgfd$, 序列 adf 是它们的一个 LCS。LCS 可能不唯一, 例如字符串 "abcd" 和 "acbd" 的 LCS 可以是 "acd" 或 "abd"。

Q02: 动态规划如何解决 LCS 问题?

- prompt: 描述动态规划在解决 LCS 问题中的递归公式, 并详细说明如何构建解法。
- model_name: ChatGPT
- model_version: GPT-4
- response: 动态规划的递归公式为:
 - i. $c[i,j]=0$ 如果 $i=0$ 或 $j=0$;
 - ii. $c[i,j]=c[i-1,j-1]+1$ 如果 $x[i]=y[j]$;
 - iii. $c[i,j]=\max\{c[i-1,j],c[i,j-1]\}$ 其他情况。该公式通过递归地比较两个序列的前缀来计算 LCS 的长度。

Q03: KMP算法如何避免文本的回溯?

- prompt: 请简要说明Knuth-Morris-Pratt算法是如何通过前缀函数避免文本回溯的。
- model_name: ChatGPT
- model_version: GPT-4
- response: KMP算法利用前缀函数 π 通过记录模式自身的重复信息来避免在文本中进行多余的比较。一旦发生配, 模式会根据 π 值跳转到新的匹配位置, 而无需回溯文本指针。

Q04: 后缀树的主要应用场景有哪些?

- prompt: 请列出后缀树在字符串处理中常见的应用场景。
- model_name: ChatGPT
- model_version: GPT-4
- response:

主要思想:

KMP 算法通过避免在文本中回溯, 提高字符串匹配的效率。其时间复杂度为 $O(n+m)$ 。

前缀函数: 前缀函数 $\pi[i]$ 表示模式串中以位置 i 结尾的前缀, 与该模式串的最长相同后缀的长度。 $\pi[i]$ 通过分析模式串自身的结构预先计算。

在匹配过程中, 当模式与文本失配时, KMP 直接利用 π 值跳转到可能的下一匹配位置, 而

不是简单地回溯文本指针。

实例：模式串 $P = \text{"ABABCABAB"}$ 的前缀函数计算为：

P : A B A B C A B A B

π : 0 0 1 2 0 1 2 3 4

例如，当匹配到位置 5 且发生失配时，直接跳转到 $\pi[5]=0$ 继续匹配，避免重复比较。

Q05：如何通过多序列比对预测蛋白质结构？

- **prompt：** 阐述多序列比对在生物信息学中如何帮助预测蛋白质的三维结构。
- **model_name：** ChatGPT
- **model_version：** GPT-4
- **response：** 多序列比对可以通过识别保守模式（motifs）来推断蛋白质的功能和结构。已知蛋白质家族成员的结构信息可以用作模板，结合比对中的保守区域，预测未知蛋白质的三维结构，因为结构比序列更具保守性。