MTH 4320 Homework 9

Yaohui Wu

April 19, 2024

Problem 1

Solution. Let $P[i]$ be the length of the longest primish subsequence $S[i]$ that ends with element $L[i]$ for $i = 1, 2, \ldots, n$. We have

$$P[1] = 1$$

$$P[2] = \begin{cases} 1 + P[1], & \text{if } S[1] + L[2] \text{ is prime} \\ 1, & \text{otherwise} \end{cases}$$

$$P[3] = \begin{cases} 1 + \max_{1 \le j < 3} P[j], & \text{if } S[j] + L[3] \text{ is prime} \\ 1, & \text{otherwise} \end{cases}$$

In general, we can deduce that

$$P[i] = \begin{cases} 1 + \max_{1 \le j < i} P[j], & \text{if } S[j] + L[i] \text{ is prime} \\ 1, & \text{otherwise} \end{cases}$$

The algorithm returns the length of the longest primish subsequence in L by computing

$$P = \max_{1 \le i \le n} P[i]$$

Assuming we can check if a number is prime in $O(1)$ time, computing $P[i]$ takes $O(i)$ time so computing P takes $O(1) + O(2) + \cdots + O(n) = O(n^2)$ time. The time complexity of the algorithm is $O(n^2)$. ■

Problem 2

Solution. Let $P(i, j)$ be the length of the longest palindrome in substring $S[i, j]$ that starts with the ith letter and ends with jth letter. We have

$$P(i,j) = \begin{cases} 1, & \text{if } i = j \\ 2 + P(i+1, j-1), & \text{if } S[i] = S[j] \\ \max\{P(i, j-1), P(i+1, j)\}, & \text{otherwise} \end{cases}$$

For $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$, we compute each $P(i, j)$ in $O(1)$ time so the time complexity of the algorithm is $O(n^2)$. ■

Problem 3

Solution. The time complexity of the algorithm is $O(n^2)$. ■