

MTH 4320 Homework 9

Yaohui Wu

April 19, 2024

Problem 1

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

$$\begin{aligned} P[1] &= 1 \\ P[2] &= \begin{cases} 1 + P[1], & \text{if } S[1] + L[2] \text{ is prime} \\ 1, & \text{otherwise} \end{cases} \\ P[3] &= \begin{cases} 1 + \max_{1 \leq j < 3} P[j], & \text{if } S[j] + L[3] \text{ is prime} \\ 1, & \text{otherwise} \end{cases} \end{aligned}$$

In general, we can deduce that

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

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

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

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

Problem 2

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

Problem 3

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