INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
CS21003 Algorithms I: First Class Test 2021 Spring

Date of Examination: 30th January 2021
Duration: 30 minutes + 5 minutes (for scanning, concatenating, and uploading)
Full Marks: 10
Subject: CS21003 Algorithms I

Part II

Answer all question.

1. What are the best-case, average-case, and worst-case time complexity of the binary search algorithm in Big-Ω notation? Justify your answer.

[2 Marks]

Solution sketch. The best-case time complexity of the binary search algorithm is $\Omega(1)$, average-case time complexity is $\Omega(\log n)$, and the worst-case time complexity is $\Omega(\log n)$. □

2. Derive the asymptotic complexity of $T(n)$ in terms of Θ for the following recurrence.

(a)

$$T(n) = \begin{cases} T(n-1) + T(n-2) & \text{if } n \geqslant 2 \\ 0 & n = 0 \\ 2 & n = 1 \end{cases}$$

[4 Marks]

Solution sketch. Prove by substitution method that $T(n) = 2F_n$ where F_n is the n-th Fibonacci number. □

(b)

$$T(n) = \begin{cases} T\left(2^{\log^{\frac{1}{11}} n}\right) + T\left(2^{\log^{\frac{9}{17}} n}\right) + 13 \log \log n & \text{if } n \geqslant 10^{20} \\ 1 & \text{otherwise} \end{cases}$$

[4 Marks]

Solution sketch. Putting $n = 2^{2^h}$, we have

$$T\left(2^{2^h}\right) = T\left(2^{2^{\frac{h}{11}}}\right) + T\left(2^{2^{\frac{9h}{17}}}\right) + 13h$$

Letting $S(h) = T\left(2^{2^h}\right)$, we have

$$S(h) = S\left(\frac{h}{11}\right) + S\left(\frac{9h}{17}\right) + 13h$$

Solving above recurrence using substitution method, we obtain

$$S(h) = \Theta(h)$$

We now have

$$T(n) = T\left(2^{2^h}\right) = S(h) = \Theta(h) = \Theta(\log \log n)$$

$\square$

All the best
