Homework 1

Deadline: 14:20 on April 2, 2024

Exercise 1.

a. (5 pts) Please order the following functions asymptotically:

$$n^{\frac{5}{6}},\ 64^{\sqrt{n}},\ n,\ \frac{\sqrt{n}}{\log^4 n},\ n!,\ \log^{50} n,\ n^{\frac{6}{5}},\ \log n!,\ 4^n$$

b. (5 pts for each) Please prove or disprove the following statements:

1. $32^{\sqrt[3]{n}} = \Omega\left(4^n\right)$.

2. $\sqrt{n}\log_2 n = o(n)$.

Exercise 2. Please analyze the time complexity of the following codes

a. (5 pts)

```
for ( int i = 1; i <= n; i = i + 1 ) {
        for ( int j = 1; j <= i^2; j = j + 1) {
                ;
        }
}
```

b. (5 pts)

```
for ( int i = 1; i <= n; i = i +1 ) {
        int j = 1;
        while ( j <= 4^i ) {
                j = j * 2;
        }
}
```

Exercise 3. Please analyze the following recursive functions asymptotically:

a. (5 pts)

$$T(n) := \begin{cases} 2 \cdot T(n/4) + T(n/2) + n & n > 1 \\ 1 & n = 1 \end{cases}$$

b. (5 pts)

$$T(n) := \begin{cases} 3 \cdot T(n/7) + \sqrt{n} & n > 1 \\ 1 & n = 1 \end{cases}$$

c. (5 pts)

$$T(n) := \begin{cases} 4 \cdot T\left(n^{\frac{1}{4}}\right) + \log_2 n & n > 4 \\ 2 & n \leq 4 \end{cases}$$

Exercise 4. (15 pts) Please analyze the expected depth of recursion for Randomized QuickSort on n distinct integers. Let $D(n)$ represent the expected depth for n numbers, and assume that $D(n)$ increases monotonically with n. (Reminder: The depth of recursion refers to the maximum number of recursive calls made before reaching the base case.)

Exercise 5. (10 pts) Given n $o(\log_2 n)^2$-bit numbers, please explain how to sort them in $o(n \log n)$ time. As in the textbook, please ignore the time of transforming a b-bit number into a $\lceil \frac{b}{r} \rceil$-digit number where $r \leq b$ is an integer.

Exercise 6. (10 pts) You have learned the *QuickSort* algorithm and the analysis of its expected time complexity using random variables. Please analyze the expected time complexity of *QuickSelect* using random variables.

Exercise 7. (10 pts) A person stands at position 0 in the real line, and he attempts to move from position 0 to position n. Moreover, some positions between 0 and n contain obstacles. The person has two moving ways:

- He can walk from position i to position $i+1$ if position $i+1$ contains no obstacle, and such a walk costs one unit of energy.
- He can jump from position i to position $i+4$ if position $i+4$ contains no obstacle, and such a jump costs six unit of energy.

Please design an algorithm that computes the minimum energy for moving from position 0 to position n, and analyze the time complexity. If a solution (or a sub-solution) does not exist, the required energy is infinitely large. (*Note*: An $w(n)$-time algorithm will not get full points.)

Exercise 8. (15 pts) A *palindrome* is a string that is exactly the same as its reversal, e.g., "madam", "level" and "rotator". The *longest palindromic subsequence* of a string is the longest subseqeuence of the string that is a palindrome. Please develop an algorithm that computes the length of the longest palindromic subsequence of an n-element string, and analyze the time complexity. (*Note*: An $w(n^2)$-time algorithm will not get full points. Please beware of the difference between a subsequence and a substring.)

Recommended Exercises:

Chapter 3 : P 3-2, P 3-3(a), P 3-4.

Chapter 4 : E 4.3-1, E 4.4-1, E 4.4-4., E 4.5-1, P 4-4.

Chapter 6 : E 6.1-8, E 6.3-4, E 6.3-4, P 6-1.

Chapter 7 : E 7.2-5, E 7.3-2, E 7.4-4, P 7-4.

Chapter 8 : E 8.2-6, E 8.3-5, E 8.4-2, P 8-2

Chapter 9 : E 9.1-2, E 9.2-3, E 9.3-3, E 9.3-6, P 9-1.