

# EL9343 Homework 1

(Due September 17th, 2019)

*All problem/exercise numbers are for the third edition of CLRS text book*

---

1. Prove the *Symmetry* property of  $\Theta(\cdot)$ , i.e.  $f(n) = \Theta(g(n))$  if and only if  $g(n) = \Theta(f(n))$ .
2. Problem 3-2 in CLRS Text book;
3. Problem 3-3 (a) in CLRS Text book;
4. First use the iteration method to solve the recurrence:

$$T(n) = 2T\left(\frac{2n}{3}\right) + n$$

then use the substitution method to verify your solution.

5. First use the iteration method to solve the recurrence

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

then use the substitution method to verify your solution.

6. Solving the recurrence:

$$T(n) = 9T(n^{\frac{1}{6}}) + \log^2(n).$$

*(Hint: Making change of variable)*

7. Give asymptotic upper and lower bounds for  $T(n)$  in each of the following recurrences. Make your bounds as tight as possible, and justify your answers.
  - (a)  $T(n) = 2T(n/3) + n^{\frac{1}{2}} \log n$
  - (b)  $T(n) = 25T(n/5) + n^2$
  - (c)  $T(n) = 4T(n/2) + n^2 \sqrt{n}$
8. Determine the time complexity of following code pieces (Figure 1), using big-O notation (every single statement has  $O(1)$  time complexity):
9. Analyze the time complexity of the following piece of code (Figure 2), using big-O notation (assume 'A' in the following code is an integer array):

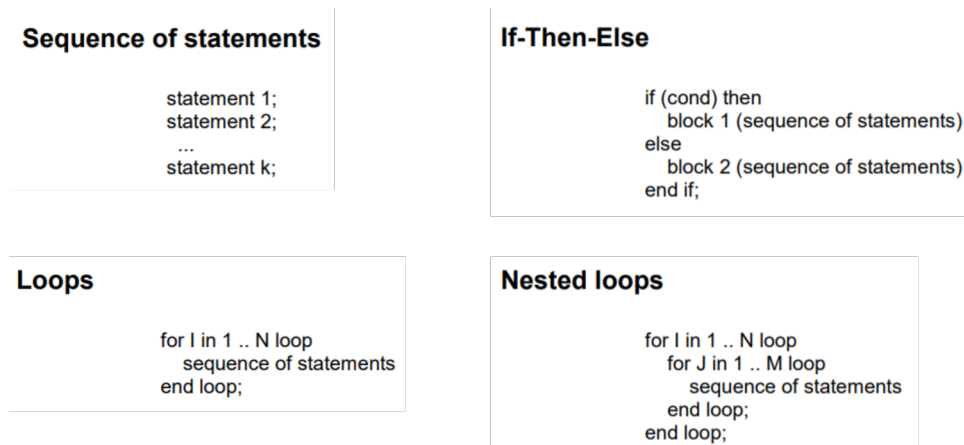


Figure 1: Code Pieces

```
low = 0
high = N - 1
while (low <= high) {
    // invariants: value > A[i] for all i < low
    //               value < A[i] for all i > high
    mid = (low + high) / 2
    if (A[mid] > value)
        high = mid - 1
    else if (A[mid] < value)
        low = mid + 1
    else
        return mid
}
```

Figure 2: Code