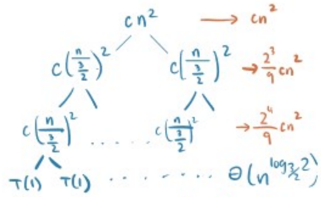
Problem 3.1

$\boxed{Q1}$ $T(n) = 2T\left(\frac{2}{3}n\right) + n^2 \quad a = 2 \quad b = \frac{3}{2}$




$\text{Total} = cn^2\left(1 + \underbrace{\frac{2^3}{9} + \frac{2^4}{9} + \frac{2^5}{9} \ldots}_{\text{geometric series = a constant}}\right)$

$= \Theta(n^2)$

---

$$< \sum_{i=0}^{\infty}\left(\frac{2^i}{9}\right)cn^2 + \Theta(n^{\log_{3/2} 2})$$

Formula

$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$

$$= \left(\frac{1}{1-\frac{2^i}{9}}\right)cn^2 + \Theta(n^{\log_{3/2} 2})$$

$$= \frac{9}{7}cn^2 + n^{1.7095}$$

$$= O(n^2) \text{ also } \Omega(n^2) \text{ meaning } \boxed{\Theta(n^2)}$$

---

Proof of master Theorem:

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ where $a \geq 1$, $b > 1$, $f(n) > 0$
$= \Theta(n^k \log^p n)$

$T(n) = 2T\left(\frac{n}{\frac{3}{2}}\right) + n^2 \quad a = 2, \; b = \frac{3}{2}, \; f(n) = n^2, \; k = 2$

compare $n^{\log_b a}$ with $f(n) = n^2$

$= n^{\log_{3/2} 2} = n^{1.7095} < f(n)$ meaning $\log_{3/2} 2 = 1.7095 < k = 2$

case 3: $\boxed{\Theta(n^2)}$

Problem 3.2

$$T(n) = 3T\left(\frac{n}{2}\right) + \frac{n^1}{\log n}$$

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ where $a \geq 1$, $b > 1$, $f(n) > 0$
$= \Theta(n^k \log^p n)$

$\log_2 3 = 1.585 > k = 1 \quad p = -1$ (log n in denominator)

case 2: if $\log_b a > k$ if $p = -1$
then $\Theta(n^{\log_b a})$

$$\boxed{= \Theta(n^{1.585})}$$

Problem 3.3.a

$\boxed{Q3.1}$

$$T(n) = \sqrt{n}\,T(\sqrt{n}) + n$$

Substitution

$$= n^{\frac{1}{2}}\,T(n^{\frac{1}{2}}) + n$$
$$= n^{\frac{1}{2}}\left[n^{\frac{1}{4}}\,T(n^{\frac{1}{4}}) + n^{\frac{1}{2}}\right] + n$$
$$= n^{\frac{1}{2}+\frac{1}{4}}\,T(n^{\frac{1}{4}}) + n^{\frac{2}{2}} + n$$
$$= n^{\frac{1}{2}+\frac{1}{4}}\,T(n^{\frac{1}{4}}) + 2n$$
$$= n^{\frac{1}{2}+\frac{1}{4}}\left[n^{\frac{1}{8}}T(n^{\frac{1}{8}}) + n^{\frac{1}{4}}\right] + 2n$$
$$= n^{\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}}\left(T(n^{\frac{1}{2^3}}\right) + n^{\frac{2}{4}+\frac{1}{4}+\frac{1}{4}} + 2n$$
$$= n^{\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}}\left(T(n^{\frac{1}{2^3}}\right) + 3n$$
$$= n^{\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}\cdots+\frac{1}{2^k}}\left(T(n^{\frac{1}{2^k}})\right) + \mathbf{K}\cdot n$$

Let $T(2) \Rightarrow n^{\frac{1}{2^k}} = 2$

$$\frac{1}{2^k}\log_2(n) = \log_2(2)$$
$$\log_2(n) = 2^k$$
$$\log_2 \log_2(n) = K \log_2(2) = K$$

Sum

$$= n^{\sum_{i=1}^{k}\frac{1}{2^i}}\left(T\left(n^{\frac{1}{2^{\log_2\log_2 n}}}\right)\right) + n\log_2\log_2(n)$$

Formula

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

$$= \frac{1}{1-\log_2 n} < n$$
$$= O(n\log_2\log_2 n)$$
$$\boxed{= \Theta(n\log\log n)}$$

Problem 3.3.b

$\boxed{Q3.2}$ $T(n) = 3T(n-1)$

Substitution

$$= 3\left[3T(n-1-1)\right]$$

$$= 3^2\ T(n-2)$$

$$= 3^2\left[3T(n-2-1)\right]$$

$$= 3^3\ T(n-3)$$

$$\vdots$$

$$= 3^n\ T(n-n)$$

$$= 3^n\ T(0)$$

$$= 3^n$$

$$\boxed{= O(3^n)} \text{ for } n > 0$$

lower bound

Problem 4 (programming assignment)

```
class Solution {
  public int maxProduct(int[] nums) {
    // check if empty array, return 0
    if (nums.length == 0) {
      return 0;
    }
    // initialize max and min tracker with nums[0] first index
    int max = nums[0];
    int min = nums[0];
    int result = max;
    // loop with nums[1] second index.
    for(int i = 1; i < nums.length; i++) {
      int currentValue = nums[i];
      // take and save the max and min found at each index, consider 0 and negatives
      int maxTemp = Math.max(currentValue, Math.max(max * currentValue, min * currentValue));
      min = Math.min(currentValue, Math.min(max * currentValue, min * currentValue));
      max = maxTemp; // assign back to max variable
      result = Math.max(max, result); // calculate result
    }
    return result;
  }
}
```