50.004 Introduction to Algorithms, Fall 2018
Singapore University of Technology and Design | 18 Sep, 2018
Gemma Roig, Ioannis Panageas, Subhajit Datta | Home Exercises - Week 1&2

Student Information

Name: ________________________ Student ID: ________________________
Due Date: 25 Sep, 11:59PM.
Submit answers on eDimension in pdf format. Submission without student information will **NOT** be marked! Any questions regarding the homework can be directed to the TA through email (contact information on eDimension).

Week 1

For all answers that are FALSE to a (T/F) question, please provide a short reason why as well.

1. The asymptotic complexity of $n^3 + 2n^2 + 1000$ is $O(n^3)$. **(T/F)**
   Solution: True

2. The asymptotic complexity of $100n^2 + n + \cos n + 1000$ is $\Theta(n^2)$. **(T/F)**
   Solution: True

3. The asymptotic complexity of $100n^{10} + n^{2.3} + 1000$ is $\Omega(n^9)$. **(T/F)**
   Solution: True

4. The asymptotic complexity of $n^2 + n + 1000$ is $\Theta(n^{1.5})$. **(T/F)**
   Solution: False, it should be $\Theta(n^2)$ instead ($\Omega(n^{1.5})$, $O(n^2)$ are acceptable as well)

5. Given a program that performs the following (assuming printing takes $\Theta(1)$):

   for(int $i = 0$; $i < n^2$; i++*)*
   for(int $j = 0$; $j < n$; j++*)*
   for(int $k = 0$; $k < 10$; k++*)*
   print(Hello)

   The asymptotic complexity is $\Theta(n^2)$. **(T/F)**
   Solution: False, the asymptotic complexity should be $\Theta(n^3)$ instead because for every iteration in the outermost for loop (n^2 iterations), n iterations will be performed. The innermost for loop can be ignored as it is consistently done 10 times.

6. Given a program that performs the following (assuming printing takes $\Theta(1)$):

   for(int $i = 0$*;* $i < 100$*;* i*++)*
   for(int $j = 0$*;* $j < n$*;* j*++)*
   print(Hello)

   The asymptotic complexity is $\Theta(n)$. **(T/F)**
   Solution: True

7. Given a program that performs the following (assuming printing takes $\Theta(1)$):

   for(int $i = 0$*;* $i < 100$*;* i*++)*
   for(int $j = 0$*;* $j < 500$*;* j*++)*
   print(n*)*

   The asymptotic complexity is $\Theta(n)$. **(T/F)**
   Solution: False, the asymptotic complexity should be $\Theta(1)$ instead as the number of iterations made by the for loops are constant with respect to n.

8. Given $f(n) = n^3 + n^2$ and $g(n) = 10n^2$, $f(n) = \Theta(g(n))$. **(T/F)**
   Solution: False, $f(n) = \Theta(n^3) \neq \Theta(g(n))$ where $g(n) = \Theta(n^2)$

9. Given $f(n) = n^{0.5} + 10$ and $g(n) = n + 10$, $f(n) = O(g(n))$. **(T/F)**
   Solution: True

10. The ranking of the functions below, sorted in **ascending** order of growth is (   ).

    A. $n^2 < nlog(n) < 2^n < n^n$
    B. $nlog(n) < n^2 < 2^n < n^n$
    C. $nlog(n) < n^2 < n^n < 2^n$
    D. $n^2 < nlog(n) < n^n < 2^n$

    Solution: B

Week 2

1) Use the Master Theorem to give tight asymptotic bounds for the following recurrences. Please show how you derive your answer.

1. $T(n) = 2T(n/4) + n^2$
   Solution: $T(n) = \Theta(n^2)$, Case 3

   $a = 2, b = 4, f(n) = n^2$
   $Height = \log_4 n, Leaves = 2^{\log_4 n} = n^{\log_4 2} = n^{0.5}$
   Since $f(n) = \Omega(n^{0.5+\epsilon})$ where $\epsilon = 1.5$
   and $a * f(n/b) = 2 * f(n/4) = 2 * (n/4)^2 = n^2/8 \leq c * f(n) = cn^2$ where $c = 1/8$ (Regularity Check)
   Case 3 applies and $T(n) = \Theta(n^2)$

2. $T(n) = 2T(4n/5) + \log n$
   Solution: $T(n) = \Theta(n^{\log_{5/4} 2})$, Case 1

   $a = 2, b = 5/4, f(n) = \log n$
   $Height = \log_{5/4} n, Leaves = 2^{\log_{5/4} n} = n^{\log_{5/4} 2} \approx n^{3.106}$
   Since $f(n) = O(n^{\log_{5/4} 2 - \epsilon})$ where $\epsilon \approx 2.106$
   Case 1 applies and $T(n) = \Theta(n^{\log_{5/4} 2})$

3. $T(n) = 2T(n/4) + \sqrt{n}$
   Solution: $T(n) = \Theta(n^{0.5} \log n)$, Case 2

   $a = 2, b = 4, f(n) = \sqrt{n}$
   $Height = \log_4 n, Leaves = 2^{\log_4 n} = n^{\log_4 2} = n^{0.5}$
   Since $f(n) = O(Leaves)$
   Case 2 applies and $T(n) = \Theta(n^{0.5} \log n)$

4. $T(n) = \sqrt{2}T(n/4) + n \log n$
   Solution: $T(n) = \Theta(n \log n)$, Case 3

   $a = \sqrt{2}, b = 4, f(n) = n \log n$
   $Height = \log_4 n, Leaves = \sqrt{2}^{\log_4 n} = n^{\log_4 \sqrt{2}} = n^{0.25}$
   Since $f(n) = \Omega(n^{0.25+\epsilon})$ where $\epsilon = 0.75$
   and $a * f(n/b) = \sqrt{2} * f(n/4) = \sqrt{2} * (n/4) \log (n/4) = (\sqrt{2}/4) n \log (n/4) \leq c * f(n) = cn \log n$ where $c = \sqrt{2}/4$ (Regularity Check)
   Case 3 applies and $T(n) = \Theta(n \log n)$