Student Information

Name: Wang Tianduo Student ID: 1002963
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)** T $n^3 = n^3$
2. The asymptotic complexity of $100n^2 + n + \cos n + 1000$ is $\Theta(n^2)$. **(T/F)** T $n^2 = n^2$
3. The asymptotic complexity of $100n^{10} + n^{2.3} + 1000$ is $\Omega(n^9)$. **(T/F)** T $n^{10} > n^9$. $\Omega(n^9)$, provide Lower boun
4. The asymptotic complexity of $n^2 + n + 1000$ is $\Theta(n^{1.5})$. **(T/F)** F $n^2 > n^{1.5}$, Should be $\Theta(n^2$
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)** F The complexity should be $\Theta(n^3)$

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)** T, two loops, one is constant length, another is n leng

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

```
for(int i = 0; i < 100; i++)
```

$$\textit{for(int } j = 0;\ j < 500;\ j{+}{+})$$
$$\textit{print}(n)$$

The asymptotic complexity is $\Theta(n)$. **(T/F)** F The complexity should be $\Theta(1)$

8. Given $f(n) = n^3 + n^2$ and $g(n) = 10n^2$, $f(n) = \Theta(g(n))$. **(T/F)** F $\rightarrow f(n) = \Omega(g(n))$
9. Given $f(n) = n^{0.5} + 10$ and $g(n) = n + 10$, $f(n) = O(g(n))$. **(T/F)** T $\because n > n^{0.5}$.
10. The ranking of the functions below, sorted in **ascending** order of growth is (B).

    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. $n^2 < nlog(n) < n^n < 2^n$

Week 2

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

1. $T(n) = 2T(n/4) + n^2$
2. $T(n) = 2T(4n/5) + \log n$
3. $T(n) = 2T(n/4) + \sqrt{n}$
4. $T(n) = \sqrt{2}T(n/4) + n \log n$

1. $a=2, b=4, f(n) = n^2$

height $= \log_4 n$. # leaves $= 2^{\log_4 n} = n^{\log_4 2} = n^{\frac{1}{2}}$

Since $f(n) = \Omega(n^{\frac{1}{2}+\varepsilon})$ where $\varepsilon = 1$,

AND

$af(\frac{n}{b}) = 2f(\frac{n}{4}) = 2(\frac{n}{4})^2 = \frac{1}{8}n^2 \leq cn^2$, for $c = \frac{1}{4}$

Case 3 can be applied and hence:

$T(n) = \Theta f(n) = \Theta(n^2)$

2. $a=2, b=\frac{5}{4}, f(n) = \log n$

height $= \log_{\frac{5}{4}} n$, # leaves $= 2^{\log_{\frac{5}{4}} n} = n^{\log_{\frac{5}{4}} 2} \approx n^{3.1}$

since $f(n) = O(n^{3.1-\varepsilon})$ where $\varepsilon = 2$,

Case 1 can be applied and hence:

$T(n) = \Theta(n^{\log_{\frac{5}{4}} 2})$.

3. $a=2, b=4.$ $f(n) = \sqrt{n} = n^{\frac{1}{2}}$

height $= \log_4 n$, # leaves $= 2^{\log_4 n} = n^{\log_4 2} = n^{\frac{1}{2}}$

Since $f(n) = \Theta(n^{\frac{1}{2}})$.

Case 2 can be applied and hence:

$T(n) = \Theta(n^{\frac{1}{2}} \log n)$

4. $a=\sqrt{2}, b=4.$ $f(n) = n\log n.$

height $= \log_4 n$, # leaves $= \sqrt{2}^{\log_4 n} = n^{\frac{1}{4}}$

Since $f(n) = \Omega(n^{\frac{1}{4}+\varepsilon})$ where $\varepsilon = \frac{3}{4}$.

AND.

$af(\frac{n}{b}) = \sqrt{2} f(\frac{n}{4}) = \frac{\sqrt{2}}{4} n \log \frac{n}{4} \le \frac{\sqrt{2}}{4} n\log n \overset{=}{\cancel{\le}} c\, f(n)$ for $c = \frac{\sqrt{2}}{4}$

$\therefore T(n) = \Theta(n\log n)$