CMPSC 465 Data Structures & Algorithms
Spring 2022 Chunhao Wang and Mingfu Shao

Solution 01

1. (0 pts.) Acknowledgements. The assignment will receive a 0 if this question is not answered.

1. If you worked in a group, list the members of the group. Otherwise, write "I did not work in a group."
2. If you received significant ideas about your solutions from anyone not in your group, list their names here. Otherwise, write "I did not consult anyone except my group members".
3. List any resources besides the course material that you consulted in order to solve the material. If you did not consult anything, write "I did not consult any non-class materials."

2. (15 pts.) For each pairs of functions below, indicate one of the three: $f = O(g)$, $f = \Omega(g)$, or $f = \Theta(g)$.

1. $f(n) = n^4$, $g(n) = (100n)^3$
2. $f(n) = n^{1.01}$, $g(n) = n^{0.99} \cdot (\log n)^2$
3. $f(n) = 4n \cdot 2^n + n^{100}$, $g(n) = 3^n$
4. $f(n) = n^2 \cdot \log(n^2)$, $g(n) = n \cdot (\log n)^3$
5. $f(n) = 3^{n-1}$, $g(n) = 3^n$
6. $f(n) = 1.01^n$, $g(n) = n^2$
7. $f(n) = 2^{\log \log n}$, $g(n) = n$
8. $f(n) = (\log n)^{100}$, $g(n) = n^{0.001}$
9. $f(n) = 5n + \sqrt{n}$, $g(n) = 3n + \log n$
10. $f(n) = 2^n + \log n$, $g(n) = 2^n + (\log n)^{10}$
11. $f(n) = \sqrt[5]{n}$, $g(n) = \sqrt[3]{n}$
12. $f(n) = n!$, $g(n) = 3^n$
13. $f(n) = \log(15n!)$, $g(n) = n \log(n^9)$
14. $f(n) = \sum_{k=1}^{n} k$, $g(n) = \log(n!)$
15. $f(n) = \sum_{k=1}^{n} k^3$, $g(n) = n^3 \cdot \log n$

Solution:

Note 1: we introduce a new definition, small ω. We define $f = \omega(g)$ if and only if $g = o(f)$. Equivalently, $f = \omega(g)$ if and only if $\lim_{n\to\infty} f(n)/g(n) = \infty$ according to the definition of small-o.

Note 2: calculating the limit of $f(n)/g(n)$ is usually an efficient approach to determine their asymptotic relationship. Specifically, $\lim_{n\to\infty} f(n)/g(n) = 0$ implies that $f = o(g)$ and consequently $f = O(g)$. Symmetrically, $\lim_{n\to\infty} f(n)/g(n) = \infty$ implies that $f = \omega(f)$ and consequently $f = \Omega(g)$. Besides, $\lim_{n\to\infty} f(n)/g(n) = c > 0$ implies that $f = \Theta(g)$.

1. $f = \Omega(g)$: $\lim_{n\to\infty} \frac{n^4}{(100n)^3} = \lim_{n\to\infty} \frac{n}{100^3} = \infty$
2. $f = \Omega(g)$: $\lim_{n\to\infty} \frac{n^{1.01}}{n^{0.99} \cdot (\log n)^2} = \lim_{n\to\infty} \frac{n^{0.02}}{(\log n)^2} = \infty$
3. $f = O(g)$: $\lim_{n\to\infty} \frac{4n \cdot 2^n + n^{100}}{3^n} = \lim_{n\to\infty} (4n \cdot \frac{2}{3}^n + \frac{n^{100}}{3^n}) = 0 + 0 = 0$

4. $f = \Omega(g)$: $\lim_{n\to\infty} \frac{n^2 \cdot \log(n^2)}{n \cdot (\log n)^3} = \lim_{n\to\infty} \frac{2n}{(\log n)^2} = \infty$
5. $f = \Theta(g)$: $\lim_{n\to\infty} \frac{3^{n-1}}{3^n} = \lim_{n\to\infty} \frac{3^n}{3 \cdot 3^n} = \frac{1}{3}$
6. $f = \Omega(g)$: an exponential function always dominates a polynomial function.
7. $f = O(g)$: $\lim_{n\to\infty} \frac{2^{\log\log n}}{n} = \lim_{n\to\infty} \frac{2^{\log\log n}}{2^{\log n}} = 0$
8. $f = O(g)$: a fractional power function always dominates a polylogarithmic function.
9. $f = \Theta(g)$: $\lim_{n\to\infty} \frac{5n+\sqrt{n}}{3n+\log n} = \lim_{n\to\infty} \frac{5n}{3n} = \frac{5}{3}$
10. $f = \Theta(g)$: $\lim_{n\to\infty} \frac{2^n+\log n}{2^n+(\log n)^{10}} = \lim_{n\to\infty} \frac{2^n}{2^n} = 1$
11. $f = O(g)$: $\lim_{n\to\infty} \frac{\sqrt[5]{n}}{\sqrt[3]{n}} = \lim_{n\to\infty} \frac{1}{n^{\frac{2}{15}}} = 0$
12. $f = \Omega(g)$: $\lim_{n\to\infty} \frac{n!}{3^n} = \lim_{n\to\infty} \frac{1 \times 2 \times \cdots n}{3 \times 3 \times \cdots \times 3} = \infty$
13. $f = \Theta(g)$: $f = \log(15n!) = \Theta(\log(n!))$. Observe that $n! = 1 \times 2 \times 3 \cdots \cdot n \leq n \cdot n \cdot n \cdots \cdot n \leq n^n$ and assuming n is even (without loss of generality) $n! = 1 \times 2 \times 3 \cdots \cdot n \geq n \cdot (n-1) \cdot (n-2) \cdots \cdot (n - n/2) \geq \left(\frac{n}{2}\right)^{\frac{n}{2}}$. Hence $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n! \leq n^n$. Then, $\frac{n}{2}\log\left(\frac{n}{2}\right) \leq \log(n!) \leq n\log n$ and $f = \Theta(\log(n!)) = \Theta(n\log n)$. $g = n\log(n^9) = 9n\log n = \Theta(n\log n)$. So, $f = \Theta(g)$.
14. $f = \Omega(g)$: First, $g = \Theta(n\log n)$ as shown above. So, $\lim_{n\to\infty} \frac{\sum_{k=1}^{n} k}{\log(n!)} = \lim_{n\to\infty} \frac{\frac{n(n+1)}{2}}{n\log n} = \lim_{n\to\infty} \frac{n}{\log n} = \infty$
15. $f = \Omega(g)$: $f = \frac{n^2(n+1)^2}{4} = \Theta(n^4)$, based on Faulhaber's formula. So, $f(n)$ dominates $g(n) = n^3 \cdot \log n$.

3. (16 pts.) Assume you have functions f, g and h. For each of the following statements, decide if you think it is true or false and give a proof or counterexample.

1. If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
2. If $f(n) = \Theta(g(n))$, then $2^{f(n)} = \Theta(2^{g(n)})$.
3. If $f(n) = o(g(n))$, then $\log f(n) = o(\log g(n))$
4. If $f(n) = O(g(n))$, then $\frac{1}{f(n)} = \Omega(\frac{1}{g(n)})$

Solution:

1. True.
   As $f(n) = O(g(n))$, there exist positive constants c_1 and N_1 such that $f(n) \leq c_1.g(n)$ for all $n \geq N_1$. Similarly, there exist positive constants c_2 and N_2 such that $g(n) \leq c_2.h(n)$ for all $n \geq N_2$. So for all n that $n \geq N_1$ and $n \geq N_2$, we have $f(n) \leq c_1.c_2.h(n)$. Replacing $c_1.c_2$ with c', we have $f(n) \leq c'.h(n)$. Therefore, $f(n) = O(h(n))$.
2. False.
   Let's consider the counterexample, f(n) = 2n and g(n) = n. f (n) is $\Theta(g(n))$ in this case because we can find constants c=2 and N =1 such that $f(n) = cg(n)$ for all $n \geq N$. However, since $2^{f(n)} = 2^{2n}$ and $2^{g(n)} = 2^n$, $\lim \frac{2^{2n}}{2^n} = \infty$. So, $2^{f(n)}$ grows faster than $2^{g(n)}$ asymptotically.Thus the statement is false.
3. False.
   A counterexample is f(n) = n and g(n) = n^2. f (n) is o(g(n)) in this case because $\lim_{n->\infty} \frac{n}{n^2} = 0$. However, $\log f(n) = \log n$ and $\log g(n) = \log n^2 = 2\log n$. As a result, $\lim_{n->\infty} \frac{\log n}{2\log n} = 1/2 \not\equiv 0$. So, the statement is false.

4. True.
As $f(n) = O(g(n))$, there exist positive constants c and N such that $f(n) \le c.g(n)$ for all $n \ge N$. Rearranging the inequality gives, $\frac{1}{f(n)} \ge \frac{1}{cg(n)} => \frac{1}{f(n)} \ge \frac{1}{c}(\frac{1}{g(n)})$. Replacing $\frac{1}{c}$ with c', we get $\frac{1}{f(n)} \ge c' \frac{1}{g(n)}$. Thus, $\frac{1}{f(n)} = \Omega(\frac{1}{g(n)})$.

4. (16 pts.) For each pseudo-code below, give the asymptotic running time in Θ notation. You may assume that standard arithmetic operations take $\Theta(1)$ time.

1.
```
for i := 1 to n do
    j := i;
    while j ≤ n do
        j := j + i;
    end
end
```

2.
```
i := 1 ;
while i ≤ n do
    j := 1;
    while j ≤ i do
        j := j + 1;
    end
    i := 2i;
end
```

3.
```
s := 0;
for i := 1 to n do
    for j := i + 1 to n do
        for k := j + 1 to n do
            s := s + 1;
        end
    end
end
```

4.
```
for i := 1 to n^2 do
    if i mod n = 0 then
        j := 1;
        while j ≤ i/n do
            j := j + 1;
        end
    end
end
```

Solution:

Note: in general, to analyze the running time of nested loops you will need to represent the running time as a summation, calculate it as a closed form, and eventually simplify it using the asymptotic notations.

Specifically, say the outer loop goes with "for $i = 1$ to n", then we want to represent the running time of inner loop as a function of i, say $F(i)$, then the entire running time will be $\sum_{i=1}^{n} F(i)$.

1. For each i, j iterates from i to n stepping by i, i.e., i, $2i$, $3i$, ..., until it reaches n. Thus, there are $\lfloor \frac{n}{i} \rfloor$ iterations for each i. Thus, the running time is

$$\sum_{i=1}^{n} \lfloor \frac{n}{i} \rfloor \approx \sum_{i=1}^{n} \frac{n}{i} = \Theta(n \log n) \tag{1}$$

2. Assume $2^k \leq n < 2^{k+1}$ for some k. Then i iterates from 1 to 2^k multiplying by 2. i.e., 1, 2, 4, 8, ..., 2^k. For each i, j iterates from 1 to i. So, the running time is

$$\sum_{l=1}^{k} \sum_{j=1}^{2^l} 1 = \sum_{l=1}^{k} 2^l = \Theta(2^k) = \Theta(n) \tag{2}$$

   Here, l is the exponent of i, namely, $i = 2^l$.

3. There is one arithmetic operation for each (i, j, k) so the running time is

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=j+1}^{n} 1 &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} (n-j) && (3)\\
&= \sum_{i=1}^{n} (\sum_{j=i+1}^{n} n - \sum_{j=i+1}^{n} j) && (4)\\
&= \sum_{i=1}^{n} n(n-i) - \frac{1}{2}(n^2 + n - i^2 - i) && (5)\\
&= \sum_{i=1}^{n} \frac{n^2}{2} - (\frac{1}{2} + i)n + \frac{i^2 + i}{2} && (6)\\
&= \frac{n^3}{2} - \frac{n^2(n+2)}{2} + \frac{n(n+1)(n+2)}{6} && (7)\\
&= \frac{1}{6} n(n^2 - 3n + 2) = \Theta(n^3) && (8)
\end{aligned}$$

4. j iterates only if i is a multiple of n. Let k be a number such that $i = kn$, that is, $k = \frac{i}{n}$. Then, j iterates from 1 to k. The running time is

$$\sum_{i=1}^{n^2} 1 + \sum_{k=1}^{n} k = n^2 + \frac{n(n+1)}{2} = \Theta(n^2) \tag{9}$$

Rubrics:

Problem 2, 15 pts

Each part has 1 point.
1 - Provide a correct answer

Problem 3, 16 pts

Each part has 4 points.
2 - Provide an appropriate proof or counterexample
2 - Provide a correct answer

Problem 4, 16 pts

Each part has 4 points.
2 - Provide an appropriate proof or justification
2 - Provide a correct answer