

I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment for my first offense and that I will receive a grade of F for the course for any additional offense.

1. Given the pseudo code below for bubble sort:

- (a) Let `length[A] = n`. What is the count for `BubbleSort(A)`? Show the steps necessary to derive your final answer.

| Line | Cost | Times |
|------|-------|--|
| 1 | c_1 | n |
| 2 | c_2 | $\sum_{i=1}^n t_i$ |
| 3 | c_3 | $\sum_{i=1}^n (t_i - 1)$ |
| 4 | c_4 | 0 (best), $\sum_{i=1}^n (t_i - 1)$ (worst) |

$$\sum_{line}^4 (\text{times line executed}) = \text{instruction count}$$

- (b) Show the worst case and best case time complexity in term of instruction counts.

Worst case scenario:

$$n + \sum_{i=1}^n t_i + \sum_{i=1}^n (t_i - 1) + \sum_{i=1}^n (t_i - 1)$$

Best case scenario:

$$n + \sum_{i=1}^n t_i + \sum_{i=1}^n (t_i - 1)$$

2. Fill in all the missing values.

| $f(n)$ | $g(n)$ | $f(n) = O(g(n))$ | $f(n) = \Omega(g(n))$ | $f(n) = \Theta(g(n))$ |
|---|-------------|------------------|-----------------------|-----------------------|
| $n^{2.125}$ | $n^2 \lg n$ | No | Yes | No |
| \sqrt{n} | n | Yes | No | No |
| $n!$ | $(n+1)!$ | Yes | No | No |
| $2^{n/2}$ | 2^n | Yes | No | No |
| $\sum_{i=1}^n i = \frac{n^2+n}{2}$ | n^2 | Yes | Yes | Yes |
| $\sum_{i=0}^{n-1} 4^i = \frac{1}{3}(4^n - 1)$ | $n4^{n-1}$ | No | Yes | No |

3. Order the functions below by increasing growth rate.

$$n^n, n \ln n, n^\epsilon (0 < \epsilon < 1), 2^{\ln n}, \ln n, 10, n!, 2^n, 10, \ln n, 2^{\ln n}, n^\epsilon (0 < \epsilon < 1), n \ln n, 2^n, n^n, n!$$

4. Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or show a counter example for each of the following conjectures.

- (a) $f(n) \in O(g(n))$ implies $2^{f(n)} \in O(2^{g(n)})$
False.

Proof. Consider $f(n) = 2n$ and $g(n) = n$. $2n \in O(n)$

Let's then assume $2^{2n} \in O(2^n)$. By definition of big-O, there must exist some constant c, n_0 such that $0 \leq 2^{2n} \leq c \cdot 2^n, \forall n \geq n_0$. Let's try to prove $2^{2n} \leq c \cdot 2^n$ for some constant c .

$$\begin{aligned} 2^{2n} &\leq c \cdot 2^n \\ (2^n)^2 &\leq c \cdot 2^n \\ 2^n &\leq \sqrt{c} \cdot 2^{n/2} \end{aligned}$$

As c is constant, as n approaches ∞ the LHS will become greater than the RHS. Therefore the implication is not true. \square

- (b) $f(n) \in O(g(n))$ implies $g(n) \in \Omega(f(n))$
True.

Proof. The definition of $f(n) \in O(g(n))$ says there must exist a constant c such that $0 \leq f(n) \leq c \cdot g(n)$.

The definition of $g(n) \in \Omega(f(n))$ says there must exist a constant c such that $0 \leq c \cdot f(n) \leq g(n)$.
Therefore $f(n) \in O(g(n)) \Leftrightarrow g(n) \in \Omega(f(n))$

\square

5. Prove $n^2 - 3n - 20 \in \Theta(n^2)$ using the original definition of Θ

Proof. By definition of Θ ,

$\Theta(n^2) = \{n^2 - 3n - 20 : \text{there exist positive constants } c_1, c_2, n_0 \text{ such that:}$
 $0 \leq c_1(n^2) \leq n^2 - 3n - 20 \leq c_2(n^2), \forall n \geq n_0\}$

Let $c_1 = \frac{1}{2}$, $c_2 = 1$ for all $n \geq 10$. First, need to show that $\frac{1}{2}n^2 \leq n^2 - 3n - 20$
Base case: $n = 10$

$$\begin{aligned}\frac{1}{2}(10)^2 &\leq (10)^2 - 3(10) - 20 \\ \frac{100}{2} &\leq 100 - 30 - 20 \\ 50 &= 50\end{aligned}$$

IH: Assume $\frac{1}{2}k^2 \leq k^2 - 3k - 20$, $k \geq 10$
Prove true for $k + 1$.

$$\begin{aligned}\frac{1}{2}(k+1)^2 &\leq (k+1)^2 - 3(k+1) - 20 \\ \frac{1}{2}(k+1)^2 &\leq k^2 + 2k + 1 - 3k - 3 - 20 \\ \frac{1}{2}(k+1)^2 &\leq k^2 - k - 22 \\ \frac{1}{2}(k+1)^2 &\leq k^2 - k - 22 - 2k + 2k + 2 - 2 \\ \frac{1}{2}(k+1)^2 &\leq k^2 - 3k - 20 + 2k - 2 \\ \frac{1}{2}(k+1)^2 &\leq \frac{1}{2}k^2 + 2k - 2 && \text{Induction step} \\ \frac{1}{2}k^2 + k + \frac{1}{2} &\leq \frac{1}{2}k^2 + 2k - 2 \\ \frac{1}{2} &< k - 2\end{aligned}$$

Now, show that $n^2 - 3n - 20 \leq n^2$
Base case: $n = 10$

$$\begin{aligned}(10)^2 - 3(10) - 20 &\leq (10)^2 \\ 100 - 30 - 20 &\leq 100 \\ 50 &\leq 100\end{aligned}$$

IH: Assume $k^2 - 3k - 20 \leq k^2$ for $k \geq 10$
 Prove true for $k + 1$.

$$\begin{aligned}
 (k+1)^2 - 3(k+1) - 20 &\leq (k+1)^2 \\
 k^2 + 2k + 1 - 3k - 3 - 20 &\leq (k+1)^2 \\
 k^2 - k - 22 &\leq (k+1)^2 \\
 k^2 - k - 22 - 2k + 2k + 2 - 2 &\leq (k+1)^2 \\
 k^2 - 3k - 20 + 2k - 2 &\leq (k+1)^2 \\
 k^2 + 2k - 2 &\leq (k+1)^2 && \text{Induction step} \\
 k^2 + 2k - 2 &\leq k^2 + 2k + 1 \\
 -2 &\leq 1
 \end{aligned}$$

This shows that $n^2 - 3n - 20 \in \Theta(n^2)$ is true with $c_1 = \frac{1}{2}$, $c_2 = 1$ for all $n \geq 10$ □

6. Disprove $n^3 \in O(n^2)$ using the original definition of O .

Proof. By definition of O ,

$O(n^2) = \{n^3 : \text{there exist positive constants } c, n_0 \text{ such that:}$
 $0 \leq n^3 \leq c(n^2), \forall n \geq n_0\}$

Need to show that $n^3 \leq c(n^2)$ is never true. Let c be any constant number.

$$\begin{aligned}
 n^3 &\leq c(n^2) \\
 1 &\leq \frac{c}{n} \\
 n &\leq c
 \end{aligned}$$

As n approaches infinity it cannot be bounded by a constant. Thus $n^3 \notin O(n^2)$. □

7. Prove $n = \omega(\lg n^2)$ using limit.

Proof. If $n = \omega(\lg n^2)$, then the limit $\lim_{n \rightarrow \infty} \frac{n}{\lg n^2}$ approaches ∞ .

$$\begin{aligned}
 \lim_{n \rightarrow \infty} \frac{n}{\lg n^2} \\
 \lim_{n \rightarrow \infty} \frac{1}{\frac{2}{n}} \\
 \lim_{n \rightarrow \infty} \frac{n}{2} \Rightarrow \infty
 \end{aligned}$$

□

8. Prove $n^a = \omega(\lg^k n)$, where $k > 0$, $a > 0$, using limit.

Proof. If $n^a = \omega(\lg^k n)$ then the limit $\lim_{n \rightarrow \infty} \frac{n^a}{\lg^k n}$ approaches ∞ .

$$\begin{aligned}
 \lim_{n \rightarrow \infty} \frac{n^a}{\lg^k n} \\
 \lim_{n \rightarrow \infty} \frac{an^a}{k \lg^{k-1}(n)}
 \end{aligned}$$

□