1. (10 points) Consider sorting \( n \) numbers stored in array \( A \) by first finding the smallest element of \( A \) and exchanging it with the number stored in \( A[1] \). Then find the second smallest element of \( A \), and exchange it with \( A[2] \). Continue in this manner for the first \( n - 1 \) elements of \( A \). This algorithm, called Selection-Sort, is given below. Assume the input and output are as specified for the sorting problem on page 16 of our textbook.

\[
\text{Selection-Sort}(A):
1. \quad n = A.\text{length}
2. \quad \text{for } j = 1 \text{ to } n - 1
3. \quad \quad \text{smallest} = j
4. \quad \quad \text{for } i = j + 1 \text{ to } n
5. \quad \quad \quad \text{if } A[i] < A[\text{smallest}]
6. \quad \quad \quad \quad \text{smallest} = i
7. \quad \quad \quad \text{exchange } A[j] \text{ with } A[\text{smallest}]
\]

(a) (1 point) Give the line number of the basic operation in the algorithm.

\text{Line 4 is the basic operation.}

(b) (2 points) Why does the outer for loop of the algorithm need to run for only the first \( n - 1 \) elements?

\text{Because if } j = n, \text{ } i \text{ (the inner for-loop counter) goes to } A[n+1], \text{ which would not be an index in the array. After the iteration in which } j = n - 1, \text{ the first } n - 1 \text{ elements have been compared and the } n - 1 \text{ smallest elements in } A \text{ are in positions } A[1] \text{ to } A[n-1]. \text{ Therefore the largest element must be in } A[n].

(c) (5 points) Are there best-case and worst-case asymptotic running times for Selection-Sort? If so, give the best-case and worst-case running times of the algorithm in \( O \) and \( \Omega \) notation. If not, give the worst-case running time in \( \Theta \) notation. Explain your answer.

\text{No, the running time is the same for all data sets of size } n \text{ because of the nested for loop and because there is no way to break out of the loops early. Only line 6 is not executed in all cases, but lines 4 and 5 are executed in every iteration. The running time is } \Theta(n^2).
2. (4 points) Use the definitions of $O$, $\Theta$, and $\Omega$ to determine whether the following assertions are true or false. Briefly justify your answers.

a. $n^2(n+1)/2 \in O(n^3)$  True  The highest term in the expression on the left is $n^3$, so we have $n^3 \leq n^3$.

b. $(n+1)/2 \in O(n)$  False  $n^2$ grows faster than $n$, so $\leq$ notation does not hold.

c. $(n+1)/2 \in \Theta(n^3)$  False  $n^2 \in O(n^3)$, but $n^2 \not\in \Omega(n^3)$, so $n^2 \not\in \Theta(n^3)$.

d. $(n+1)/2 \in \Omega(n)$  True  $n^2$ grows faster than $n$, so $\geq$ notation holds.

3. (4 points) For each of the following functions, indicate the class $\Theta(g(n))$ the function belongs to. Show your work in simplifying these expressions to arrive at each answer.

a. $(n^2+1)^{10} \approx (n^{20}+1) \in \Theta(n^{20})$

b. $\sqrt{10n^2+7n+3} = (10n^2+7n+3)^{\frac{1}{2}} = (10n+7n^{\frac{1}{2}}+3^{\frac{1}{2}}) \in \Theta(n)$

c. $2^{n+1} + 3^{n-1} \in \Theta(3^{n-1}) \forall \ n \geq 5$

d. $|\log n| \leq (\log(2n) - 1) = \log(n) + \log(2) - 1 = \log(n) + 1 - 1 \in \Theta(\log n)$.

(Robert Elizardo, S19) Given $c_1 = \frac{1}{2}, c_2 = 1$, and any $n_0$, $c_1 g(\log n) \leq |\log n| \leq c_2 g(\log n)$. Therefore, $T(n) \in \Theta(\log n)$.

(Ross Guju, S19) We know $x - 1 < |x| \leq x$, so we see $\log n - 1 < |\log n| \leq \log n$. Thus we have our upper bound $|\log n| \leq \log n$.

Similarly, for the lower bound we see: $|\log n| > \log n - 1 \geq \log n - \frac{1}{2} \log n = \frac{1}{2} \log n$. Then we have the boundaries $\frac{1}{2} \log n < |\log n| \leq \log n$. Hence $|\log n| \in \Theta(\log n)$.

(Corin Rose, S19) We can see that $|\log n|$ can be bounded above by $\log n$ and below by $\frac{1}{2} \log n$, and so $|\log n| \in \Theta(\log n)$.

4. (4 points) Answer Yes or No and provide a brief justification for each question below:

a. Is $2^{n+1} \in O(2^n)$? Yes, because $\lim_{n \to \infty} \frac{2^{n+1}}{2^n} = 2$, so $2^{n+1} \in \Theta(2^n)$ and therefore $2^{n+1} \in O(2^n)$.

b. Is $2^n \in O(2^n)$? No, because $\lim_{n \to \infty} \frac{2^n}{2^n} = 2^n = \infty$, which means that $2^n \in \omega(2^n)$.