

Algorithms Exercise Sheet: Big-O Reasoning

Instructions

Answer the following questions to deepen your understanding of big-O notation. Show all work for full credit.

Exercises

- 1. Determine the Big-O classification:** For each of the following functions, determine the tightest big-O classification:

a. $f(n) = 3n^2 + 7n + 10$

b. $f(n) = 5n \log n + n^2$

c. $f(n) = 2^n + n^3$

- 2. Comparing growth rates:** Rank the following functions by growth rate from slowest to fastest. Justify your rankings.

a. n^2

b. $n \log n$

c. 2^n

d. $n!$

- 3. Big-O proofs:** Prove that $n^3 + 100n^2 + 50n + 10 = O(n^3)$.

- 4. Analyzing pseudocode: Selection Sort**

The following pseudocode describes the Selection Sort algorithm:

```
Algorithm SelectionSort(A[0..n-1]):  
  for i = 0 to n - 2:  
    minIndex = i  
    for j = i + 1 to n - 1:  
      if A[j] < A[minIndex]:  
        minIndex = j  
    swap A[i] with A[minIndex]
```

Determine the order of growth (big-O) of the best-case and worst-case running times of this algorithm.

5. Analyzing pseudocode: Insertion Sort

The following pseudocode describes the Insertion Sort algorithm:

```
Algorithm InsertionSort(A[0..n-1]):  
for i = 1 to n - 1:  
    key = A[i]  
    j = i - 1  
    while j >= 0 and A[j] > key:  
        A[j + 1] = A[j]  
        j = j - 1  
    A[j + 1] = key
```

Determine the order of growth (big-O) of the best-case and worst-case running times of this algorithm.