Assignment 3: Code Complexity Analysis

SCIS 313: Data Structures and Algorithm Analysis

Prof. Antonio Khalil Moretti
Spelman College

Due: [Insert Date]

Instructions

For each problem below, analyze the given code snippet and answer the following questions:

(a) How many times does the outermost loop execute? (Express in terms of n)

(b) If there are nested loops, how many times does each inner loop execute?

(c) What is the total number of times the loop body executes? (Express as $T(n)$)

(d) What is the Big-O complexity? Simplify to the tightest bound.

Show your work! Write out the step-by-step analysis as demonstrated in class.

Problems

Problem 1: Warm-up

```
int result = 0;
for (int i = 0; i < 3 * n; i++) {
    result += i;
}
```

Problem 2: Nested Loops

```
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < 5; j++) {
        sum += i * j;
    }
}
```

Problem 3: Triangular Pattern

```
int count = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j <= i; j++) {
        count++;
    }
}
```

Hint: This forms a triangular pattern. Remember: $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$

Problem 4: Halving Loop

```
int k = n;
while (k > 0) {
    // do some O(1) work
    k = k / 3;
}
```

Hint: Each iteration divides by 3. How many times can you divide n by 3 before reaching 1?

Problem 5: Sequential Sections

```
// Section 1
for (int i = 0; i < n; i++) {
    sum += i;
}

// Section 2
for (int i = 1; i < n; i = i * 2) {
    sum += i;
}

// Section 3
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        sum += i + j;
    }
}
```

Hint: Analyze each section separately, then combine using the rule for sequential code.

Problem 6: Nested with Doubling

```
int total = 0;
for (int i = 0; i < n; i++) {
    for (int j = 1; j < n; j = j * 2) {
        total += i + j;
    }
}
```

Problem 7: Reverse Triangular

```
int product = 1;
for (int i = 0; i < n; i++) {
    for (int j = i; j < n; j++) {
        product *= 2;
    }
}
```

Hint: When $i = 0$, inner loop runs n times. When $i = 1$, inner runs $n - 1$ times, etc.

Problem 8: Triple Nesting

```
int value = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        for (int k = 0; k < n; k++) {
            value++;
        }
    }
}
```

Problem 9: Different Bounds

```
int x = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        for (int k = 0; k < n; k++) {
            x += i + j + k;
        }
    }
}
```

Note: Express your answer in terms of both n and m.

Problem 10: Challenge Problem

```
int total = 0;
for (int i = 1; i < n; i = i * 2) {
    for (int j = 0; j < i; j++) {
        total += j;
    }
}
```

Hint: The outer loop runs $\log n$ times, but the inner loop size changes! When $i = 1$, inner runs 1 time. When $i = 2$, inner runs 2 times. When $i = 4$, inner runs 4 times, etc. This is a geometric series.

Submission Guidelines

- Submit a PDF document with your solutions
- For each problem, clearly label parts (a), (b), (c), and (d)
- Show all mathematical work and reasoning
- Use proper Big-O notation: $O(n)$, $O(n^2)$, $O(\log n)$, etc.
- If you're unsure, explain your reasoning — partial credit is available!

Grading Rubric

Each problem is worth 10 points:

- 2 points: Correct outer loop count
- 2 points: Correct inner loop analysis (if applicable)

- 3 points: Correct total operation count $T(n)$
- 2 points: Correct Big-O complexity
- 1 point: Clear explanation and work shown

Total: 100 points