

(I) Give the tightest bound in big -O notation for the following code snippets:

```
(a) int sum = 0;
    for (int i = 1; i <= N + 2; i++) {
        sum++;
    }
    for (int j = 1; j <= N * 2; j++) {
        sum++;
    }
    cout << sum << endl;

(b) int sum = 0;
    for (int i = 1; i <= N - 5; i++) {
        for (int j = 1; j <= N - 5; j += 2) {
            sum++;
        }
    }
    cout << sum << endl;

(c) int sum = 0;
    for (int i = 0; i < 1000000; i++) {
        for (int j = 1; j <= i; j++) {
            sum += N;
        }
        for (int j = 1; j <= i; j++) {
            sum += N;
        }
        for (int j = 1; j <= i; j++) {
            sum += N;
        }
    }
    cout << sum << endl;
```

(II) Determine the big-O runtime of each of the following functions , in terms of the variable n :

```
(a) void function1(int n) {
    for (int i = 0; i < n; i++) {
        cout << '*' << endl;
    }
}

(b) void function2(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << '*' << endl;
        }
    }
}
```

```
(c) void function3(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            cout << '*' << endl;
        }
    }
}
```

```
(d) void function4(int n) {
    for (int i = 1; i <= n; i *= 2) {
        cout << '*' << endl;
    }
}
```

(III) Exercises in Algorithms Illuminated part 1: The basics

- (a) Problem 1.1 page 33
- (b) Problem 1.2 page 33
- (c) Problem 1.3 page 34
- (d) Problem 1.4 page 34
- (e) Problem 1.5 page 35
- (f) Problem 2.1 page 57
- (g) Problem 2.2 page 58
- (h) Problem 2.3 page 58
- (i) Problem 2.4 page 58
- (j) Problem 2.5 page 59
- (k) Problem 3.1 page 90
- (l) Problem 3.2 page 91
- (m) Problem 3.3 page 91
- (n) Problem 4.1 page 114
- (o) Problem 4.2 page 115
- (p) Problem 4.3 page 115
- (q) Problem 4.4 page 115