

# QB2

## Big Oh with logs and roots

Do NOT write on this test. Record all answers on the bubble sheet. **Closed book. No notes.** Work strictly from memory. No time limit. **No calculators. Scratch paper okay.**

1/2p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^9$  (C)  $n^7$  (E)  $n^5$  (G)  $n^3$  (I)  $n$   
(B)  $n^8$  (D)  $n^6$  (F)  $n^4$  (H)  $n^2$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        g = n; while ( g > 1 ) {
            if ( simpleCompare ) {
                simpleStatement;
            }
            g -= 3; }
    } else {
        i = 1; while ( i < n ) {
            simpleStatement;
            i++; }
    }
    return 0; }
```

---

2/2p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^3$  (C)  $n^2\sqrt{n}$  (E)  $n\sqrt{n}\lg n$  (G)  $n\lg^3 n$  (I)  $\lg^2 n$   
(B)  $n^2\sqrt{n}\lg n$  (D)  $n^2$  (F)  $n\sqrt{n}$  (H)  $\sqrt{n}\lg n$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    for ( b = 1 ; b < n ; b *= 2 ) {
        if ( simpleCompare ) {
            for ( i = n ; i > 1 ; i /= 2 ) {
                for ( j = n ; j > 1 ; j /= 5 ) {
                    for ( d = n ; d > 1 ; d -= 2 ) {
                        simpleStatement;
                    }
                }
            }
        } else {
            if ( simpleCompare ) {
                for ( e = n ; e > 1 ; e /= 2 ) {
                    simpleStatement;
                }
            }
        }
    }
    return 0; }
```

# QB2

3/2p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program. Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^5\sqrt{n}$  (C)  $n^3\sqrt{n}\lg n$  (E)  $n^3\lg^3 n$  (G)  $n^2\lg^4 n$  (I)  $n\sqrt{n}\lg^3 n$   
(B)  $n^5$  (D)  $n^3\sqrt{n}$  (F)  $n^2\sqrt{n}\lg^2 n$  (H)  $n^2$  (J)  $n$

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    i = n; while ( i > 1 ) {
        if ( simpleCompare ) {
            h = 1; while ( h * h < n ) {
                if ( simpleCompare ) {
                    for ( f = n ; f > 1 ; f -= 1 ) {
                        b = 1; while ( b < n ) {
                            for ( e = n ; e > 1 ; e -= 1 ) {
                                if ( simpleCompare ) {
                                    for ( a = n ; a > 1 ; a -= 1 ) {
                                        simpleStatement;
                                    }
                                }
                            }
                        }
                    }
                }
                b++; }
            }
        h++; }
    } else {
        if ( simpleCompare ) {
            if ( simpleCompare ) {
                d = n; while ( d > 1 ) {
                    for ( k = 1 ; k * k < n ; k += 5 ) {
                        if ( simpleCompare ) {
                            simpleStatement;
                        } else {
                            simpleStatement;
                        }
                    }
                }
                d /= 2; }
            }
        }
        i--; }
    return 0; }
```

# QB2

4/2p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.  
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^3$       (C)  $n^2 \lg n$     (E)  $n\sqrt{n} \lg n$     (G)  $n$       (I)  $\sqrt{n}$   
(B)  $n^2\sqrt{n}$     (D)  $n^2$       (F)  $n \lg^2 n$       (H)  $\sqrt{n} \lg n$     (J)  $\lg n$

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        if ( simpleCompare ) {
            h = 1; do {
                if ( simpleCompare ) {
                    d = n; while ( d > 1 ) {
                        for ( k = n ; k > 1 ; k /= 2 ) {
                            simpleStatement;
                        }
                        d /= 2; }
                    }
                h++; } while ( h * h < n );
        } else {
            if ( simpleCompare ) {
                c = 1; do {
                    simpleStatement;
                } while ( c < n );
            }
        }
    } else {
        f = n; while ( f > 1 ) {
            if ( simpleCompare ) {
                if ( simpleCompare ) {
                    g = 1; do {
                        for ( j = n ; j > 1 ; j /= 2 ) {
                            simpleStatement;
                        }
                        g++; } while ( g < n );
                    } else {
                        simpleStatement;
                    }
                }
            }
            f /= 3; }
        }
    return 0; }
```

# QB2

5/2p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.  
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^2\sqrt{n}$  (C)  $n^2$  (E)  $n \lg n$  (G)  $\sqrt{n} \lg n$  (I)  $\lg^2 n$   
(B)  $n^2 \lg n$  (D)  $n\sqrt{n}$  (F)  $n$  (H)  $\sqrt{n}$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        if ( simpleCompare ) {
            if ( simpleCompare ) {
                if ( simpleCompare ) {
                    b = 1; do {
                        for ( h = n ; h > 1 ; h-- ) {
                            if ( simpleCompare ) {
                                simpleStatement;
                            } else {
                                simpleStatement;
                            }
                        }
                    } while ( b < n );
                } else {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            simpleStatement;
                        }
                    }
                }
            } else {
                for ( i = 1 ; i < n ; i *= 2 ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            if ( simpleCompare ) {
                                simpleStatement;
                            }
                        } else {
                            simpleStatement;
                        }
                    } else {
                        simpleStatement;
                    }
                }
            }
        }
    }
    return 0; }
```

# QB2

Bro Colton

CS 201 Big Oh with logs and roots

5

---

Total points 10.

# QB2

**Answer Key** (points per line)

- |        |                      |
|--------|----------------------|
| 1 (2). | I ( $n$ )            |
| 2 (2). | G ( $n \lg^3 n$ )    |
| 3 (2). | A ( $n^5 \sqrt{n}$ ) |
| 4 (2). | F ( $n \lg^2 n$ )    |
| 5 (2). | E ( $n \lg n$ )      |

Total points 10.