

# 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 ) {
                for ( i = n ; i > 1 ; i -= 10 ) {
                    simpleStatement;
                }
            } else {
                simpleStatement;
            }
            g -= 1; }
    } else {
        if ( simpleCompare ) {
            simpleStatement;
        }
    }
    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)  $n^2\sqrt{n}$  (C)  $n\sqrt{n}\lg n$  (E)  $n\lg n$  (G)  $\sqrt{n}\lg n$  (I)  $\lg n$   
(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 ) {
            b = n; while ( b > 1 ) {
                simpleStatement;
                b -= 2; }
        } else {
            simpleStatement;
        }
    } else {
        e = 1; do {
            simpleStatement;
            e *= 3; } while ( e < n );
    }
    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)  $n^4\sqrt{n}\lg n$  (C)  $n^4\lg^2 n$  (E)  $n^3\lg^2 n$  (G)  $n^2\lg n$  (I)  $\lg n$   
(B)  $n^4\lg^3 n$  (D)  $n^3\sqrt{n}$  (F)  $n^3\lg n$  (H)  $n$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    i = 1; do {
        k = n; do {
            d = n; while ( d > 1 ) {
                for ( f = 1 ; f * f < n ; f++ ) {
                    e = 1; while ( e < n ) {
                        if ( simpleCompare ) {
                            if ( simpleCompare ) {
                                if ( simpleCompare ) {
                                    j = 1; do {
                                        if ( simpleCompare ) {
                                            if ( simpleCompare ) {
                                                a = n; do {
                                                    simpleStatement;
                                                    a /= 3; } while ( a > 1 );
                                                }
                                            }
                                        }
                                    j *= 2; } while ( j < n );
                                }
                            }
                        } else {
                            if ( simpleCompare ) {
                                if ( simpleCompare ) {
                                    b = n; do {
                                        simpleStatement;
                                        b--; } while ( b > 1 );
                                    }
                                } else {
                                    if ( simpleCompare ) {
                                        simpleStatement;
                                    }
                                }
                            }
                        }
                    e *= 2; }
                }
            d--; }
        k /= 2; } while ( k > 1 );
    i++; } while ( i * i < n );
    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)  $n^2\sqrt{n}$  (C)  $n\sqrt{n}\lg n$  (E)  $n\lg^2 n$  (G)  $\sqrt{n}$  (I)  $\lg^2 n$   
(B)  $n^2\lg n$  (D)  $n\sqrt{n}$  (F)  $n\lg n$  (H)  $\lg^3 n$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        c = n; while ( c > 1 ) {
            g = 1; while ( g * g < n ) {
                if ( simpleCompare ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            if ( simpleCompare ) {
                                if ( simpleCompare ) {
                                    if ( simpleCompare ) {
                                        j = 1; while ( j * j < n ) {
                                            simpleStatement;
                                            j++; }
                                        }
                                    } else {
                                        if ( simpleCompare ) {
                                            simpleStatement;
                                        } else {
                                            simpleStatement;
                                        }
                                    }
                                } else {
                                    d = 1; do {
                                        if ( simpleCompare ) {
                                            simpleStatement;
                                        } else {
                                            simpleStatement;
                                        }
                                        d++; } while ( d * d < n );
                                }
                            }
                        }
                    }
                }
            }
        }
        g++; }
    c /= 5; }
}
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)  $n^3\sqrt{n}$  (C)  $n^2\lg^2 n$  (E)  $n\sqrt{n}\lg^2 n$  (G)  $n$  (I)  $\lg^4 n$   
(B)  $n^3\lg n$  (D)  $n^2\lg n$  (F)  $n\sqrt{n}\lg n$  (H)  $\sqrt{n}$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    j = 1; do {
        if ( simpleCompare ) {
            for ( e = 1 ; e < n ; e *= 2 ) {
                for ( i = n ; i > 1 ; i /= 3 ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            f = 1; while ( f < n ) {
                                if ( simpleCompare ) {
                                    simpleStatement;
                                } else {
                                    simpleStatement;
                                }
                            }
                            f *= 2; }
                        }
                    } else {
                        if ( simpleCompare ) {
                            simpleStatement;
                        }
                    }
                }
            }
        } else {
            for ( d = 1 ; d < n ; d *= 2 ) {
                if ( simpleCompare ) {
                    g = 1; do {
                        a = 1; do {
                            simpleStatement;
                            a++; } while ( a < n );
                        g += 10; } while ( g * g < n );
                } else {
                    h = n; do {
                        simpleStatement;
                        h /= 2; } while ( h > 1 );
                }
            }
        }
        j *= 5; } while ( j < n );
    return 0; }
```

# QB2

Bro Colton

CS 201 Big Oh with logs and roots

5

---

Total points 10.

q10 v1115959573

# QB2

**Answer Key** (points per line)

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

Total points 10.