

# 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 ) {
        if ( simpleCompare ) {
            simpleStatement;
        } else {
            simpleStatement;
        }
    } else {
        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)  $n^2$  (E)  $n\sqrt{n}$  (G)  $\sqrt{n}\lg n$  (I)  $\lg n$   
(B)  $n^2\lg n$  (D)  $n\sqrt{n}\lg n$  (F)  $n$  (H)  $\lg^2 n$  (J) 1

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

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    for ( f = 1 ; f * f < n ; f++ ) {
        if ( simpleCompare ) {
            if ( simpleCompare ) {
                b = 1; do {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            for ( k = n ; k > 1 ; k /= 2 ) {
                                simpleStatement;
                            }
                        }
                    } else {
                        for ( e = 1 ; e * e < n ; e += 3 ) {
                            simpleStatement;
                        }
                    }
                } while ( b < n );
            }
        } else {
            for ( h = n ; h > 1 ; h -= 5 ) {
                if ( simpleCompare ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            simpleStatement;
                        } else {
                            simpleStatement;
                        }
                    } else {
                        simpleStatement;
                    }
                }
            }
        }
    }
    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)  $n$  (I)  $\sqrt{n}\lg n$   
(B)  $n^2\lg n$  (D)  $n\sqrt{n}$  (F)  $n\lg n$  (H)  $\sqrt{n}\lg^2 n$  (J)  $\sqrt{n}$

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

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    e = 1; do {
        if ( simpleCompare ) {
            if ( simpleCompare ) {
                if ( simpleCompare ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            if ( simpleCompare ) {
                                simpleStatement;
                            }
                        }
                    }
                }
            } else {
                b = 1; do {
                    if ( simpleCompare ) {
                        simpleStatement;
                    }
                } while ( b * b < n );
            }
        } else {
            if ( simpleCompare ) {
                d = n; while ( d > 1 ) {
                    if ( simpleCompare ) {
                        a = n; do {
                            simpleStatement;
                            a /= 2; } while ( a > 1 );
                        }
                    d /= 5; }
                }
            }
        }
    } while ( e < n );
    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).	J (1)
2 (2).	F ( $n$ )
3 (2).	E ( $n\sqrt{n}$ )
4 (2).	A ( $n^2\sqrt{n}$ )
5 (2).	H ( $\sqrt{n} \lg n$ )

Total points 10.