

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

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        j = 1; do {
            d = 1; while ( d < n ) {
                for ( a = n ; a > 1 ; a -= 1 ) {
                    simpleStatement;
                }
                d++; }
            j += 10; } while ( j * j < n );
    } else {
        for ( c = 1 ; c < n ; c++ ) {
            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) $n\sqrt{n}$ (E) n (G) \sqrt{n} (I) $\lg n$
 (B) n^2 (D) $n\lg n$ (F) $\sqrt{n}\lg n$ (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        if ( simpleCompare ) {
            for ( g = 1 ; g < n ; g += 3 ) {
                if ( simpleCompare ) {
                    simpleStatement;
                } else {
                    simpleStatement;
                }
            }
        } else {
            i = 1; while ( i * i < n ) {
                simpleStatement;
                i += 10; }
        }
    }
    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) $n^2\sqrt{n}$ (C) $n\sqrt{n}$ (E) n (G) \sqrt{n} (I) $\lg n$
(B) $n^2 \lg n$ (D) $n \lg n$ (F) $\sqrt{n} \lg n$ (H) $\lg^2 n$ (J) 1

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

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        i = 1; do {
            if ( simpleCompare ) {
                d = 1; while ( d < n ) {
                    b = n; while ( b > 1 ) {
                        j = 1; do {
                            simpleStatement;
                            j++; } while ( j * j < n );
                        b -= 10; }
                    d += 1; }
            } else {
                if ( simpleCompare ) {
                    if ( simpleCompare ) {
                        simpleStatement;
                    }
                } else {
                    simpleStatement;
                }
            }
            i++; } while ( i * i < n );
    } else {
        for ( f = 1 ; f < n ; f++ ) {
            if ( simpleCompare ) {
                if ( simpleCompare ) {
                    if ( simpleCompare ) {
                        if ( simpleCompare ) {
                            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). | A ($n^2\sqrt{n}$) |
| 2 (2). | E (n) |
| 3 (2). | E (n) |
| 4 (2). | F (n) |
| 5 (2). | C (n^3) |

Total points 10.