

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

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

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

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

Total points 10.