

CSE 373 Section 2: Asymptotic Analysis

Big O Proofs

(1) For each of the following, show that $f \in O(g)$. That is you will need to find values for c and n_0 such that the definition of Big-O holds true.

a) $f(n) = 7n$

$$g(n) = \frac{n}{10}$$

b) $f(n) = 1000$

$$g(n) = 3n^3$$

c) $f(n) = 7n^2 + 3n$

$$g(n) = n^4$$

d) $f(n) = n + 2n \lg(n)$

$$g(n) = n \lg(n)$$

Asymptotic Analysis

(2) Order these functions from fastest to slowest in terms of asymptotic runtime

- a) $N(N^2 \log(N) + N)$
- b) N^2
- c) $10,000N^3$
- d) $2^N + 3.14159$
- e) $N^{\frac{1}{2}} + N + 128$

(3) For each of the following, determine the tight $O(-)$ bound for the worst-case runtime in terms of the free variables of the code snippets

a)

```
int x = 0;
for (int i = n; i >= 0; i--) {
    if ((i \% 3) == 0) {
        break
    } else {
        x += n
    }
}
```

b)

```
int x = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < (n * n / 3); j++) {
        x += j;
    }
}
```

c)

```
int x = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < i; j++) {
        x += j;
    }
}
```

d)

```
int x = 0;
for (int i = 0; i < n; i++) {
    if( n < 100000) {
        for (int j = 0; j < i * i * n; j++) {
            x += j;
        }
    } else {
        x += 1 }
}
```