**2.1**

**9.** For each of the following pairs of functions, indicate whether the first function of each of the following pairs has a lower, same, or higher order of growth (to within a constant multiple) than the second function.

**a.** $n(n+1)$ and $2000n^2$ **b.** $100n^2$ and $0.01n^3$

**c.** $\log_2 n$ and $\ln n$ **d.** $\log_2^2 n$ and $\log_2 n^2$

**e.** $2^{n-1}$ and $2^n$ **f.** $(n-1)!$ and $n!$

a) $n(n+1) = n^2 + n$

$2000n^2$

These functions are the same order of growth

b) $100n^2$

$0.001n^3$ ✓

c) $\log_2 n = \frac{\log n}{\log 2}$

$\ln n = \frac{\log a}{\log e}$

$$\lim_{n \to \infty} \frac{\frac{\cancel{\log n}}{\log 2}}{\frac{\cancel{\log n}}{\log e}} = \frac{\log e}{\log 2}$$

$\simeq c$

d) $\log_2^2 n = \log_2 n \cdot \log_2 n$

$\log_2 n^2 = 2\log_2 n$

$$\lim_{n \to \infty} \frac{\log_2 n \cdot \cancel{\log_2 n}}{2\cancel{\log_2 n}} = \frac{\log_2 n}{2}$$

the $\log_2^2 n$ has great growth

e) $\frac{2^{n-1}}{2^n} = \frac{2^n}{2}$ ⟩ these have same order of growth

f) $\frac{(n-1)!}{(n)!} = n.(n-1)!$ → has higher order of growth

**2.2**

**2.** Use the informal definitions of $O$, $\Theta$, and $\Omega$ to determine whether the following assertions are true or false.

**a.** $n(n+1)/2 \in O(n^3)$ **b.** $n(n+1)/2 \in O(n^2)$
**c.** $n(n+1)/2 \in \Theta(n^3)$ **d.** $n(n+1)/2 \in \Omega(n)$

a) $\frac{n(n+1)}{2} \overset{?}{\in} O(n^3)$ ✗

$\in O(n^2)$

b) $\frac{n(n+1)}{2} \overset{?}{\in} O(n^2)$ ✓

c) $\frac{n(n+1)}{2} \in \Theta(n^3)$ ✗

?

d) $\frac{n(n+1)}{2} \in \Omega(n)$ True

**3.** For each of the following functions, indicate the class $\Theta(g(n))$ the function belongs to. (Use the simplest $g(n)$ possible in your answers.) Prove your assertions.

**a.** $(n^2+1)^{10}$ **b.** $\sqrt{10n^2+7n+3}$

**c.** $2n\lg(n+2)^2 + (n+2)^2\lg\frac{n}{2}$ **d.** $2^{n+1}+3^{n-1}$

**e.** $\lfloor \log_2 n \rfloor$

a) $(n^2+1)^{10}$

$(n^2+1)^{10} \approx (n^2)^{10} = n^{20} \in \Theta(n^{20})$

b) $\sqrt{10n^2+7n+3} \approx \sqrt{10n^2} = \sqrt{10}n \in \Theta(n)$

c) $2n\lg(n+2)^2 + (n+2)^2\lg\frac{n}{2}$

$\underbrace{2 \cdot 2n\lg(n+2)}_{\Theta(n\log n)} + \underbrace{(n+2)^2\lg\frac{n}{2}}_{\Theta(n^2 \cdot \log n)} = \Theta(n^2\log n)$

d) $2^{n+1}+3^{n-1} \in \Theta(\cdot)$

$\underbrace{2 \cdot 2^n}_{\Theta(2^n)} + \underbrace{\frac{3^n}{3}}_{\Theta(3^n)} = \Theta(3^n)$

e) $\lfloor \log_2 n \rfloor$

$\lfloor \log_2 n \rfloor \approx \log_2 n \quad \Theta(\log_2 n)$

$x-1 < \lfloor x \rfloor \leq x$

**5.** List the following functions according to their order of growth from the lowest to the highest:

$$(n-2)!,\quad 5\lg(n+100)^{10},\quad 2^{2n},\quad 0.001n^4+3n^3+1,\quad \ln^2 n,\quad \sqrt[3]{n},\quad 3^n.$$

$(n-2)! \quad \Theta((n-2)!)$

$5\log(n+100)^{10} \quad \Theta(\log n)$

$2^{2n} \quad \Theta(4^n)$

$4^n$

$(2^2)^n = 4^n$

$0{,}001n^4 + \quad \Theta(n^4)$

$\ln^2 n \Rightarrow \ln n . \ln n \quad \Theta(\ln^2 n)$

$\sqrt[3]{n} = n^{1/3} \quad \Theta(n^{1/3}) = \Theta(\sqrt[3]{n})$

$3^n \quad \Theta(3^n)$

$0{,}001n^4+3n^3+1$

$5\lg(n+200)^{\ldots}$, $\ln^{\ldots} n$, $\sqrt{n}$, ...

$3^n$, $2^{2n}$, $(n-2)!$

**2.3**

2. Find the order of growth of the following sums. Use the $\Theta(g(n))$ notation with the simplest function $g(n)$ possible.
   - **a.** $\sum_{i=0}^{n-1}(i^2+1)^2$
   - **b.** $\sum_{i=2}^{n-1} \lg i^2$
   - **c.** $\sum_{i=1}^{n}(i+1)2^{i-1}$
   - **d.** $\sum_{i=0}^{n-1}\sum_{j=0}^{i-1}(i+j)$

a) $\sum_{i=0}^{n-1}(i^2+1)^2 = \sum_{i=0}^{n-1} i^4+2i^2+1$

$\sum_{i=0}^{n-1} i^4 + \sum_{i=0}^{n-1} 2i^2 + \sum_{i=0}^{n-1} 1$

$\hookrightarrow \frac{n(n+1)(6n^3+9n^2+n-1)}{30} \approx n^5 \in \Theta(n^5)$

$\sum_{i=0}^{n-1} 2i^2 = \frac{n(n+1)}{2} \approx n^2 \in \Theta(n^2)$

$\sum_{i=0}^{n-1} i = n-1-0-1 = n \in \Theta(n)$

$\Theta(n^5)+\Theta(n^2)+\Theta(n) = \Theta(n^5)$

$\sum^{n-1} \log i = 2\sum^{n} \lg i -$

b) $\sum_{i=2}^{n-1} \log_2 i^2 = 2 \sum_{i=2} \ldots 2$ $\quad i=1 \quad 2\log_2 n$

$= 2 \underbrace{\sum_{i=1}^{n} \log_2 i} - 2\log_2 n \rightarrow \Theta(\lg n)$

$= \underbrace{\log_2 1 + \log_2 2 + \log_2 3 + \ldots + \log_2 n}_{n \text{ time}}$

$= n \cdot \log_2 n \in \Theta(n\lg n)$

$= \Theta(n\lg n) + \Theta(\lg n) = \Theta(n\lg n)$

c) $\sum_{i=1}^{n} (i+1) 2^{i-1} = \sum_{i=1}^{n} \left(2^{i-1} i + 2^{i-1}\right)$

$\sum_{i=1}^{n} i \cdot 2^{i-1} + \sum_{i=1}^{n} 2^{i-1} \rightarrow \Theta(2^n)$

$\hookrightarrow \sum_{i=1} i \cdot 2^{i-1} = 1 \cdot 2^0 + 2 \cdot 2 + 3 \cdot 2^2 \ldots$

$= \frac{2^{n-1}}{n-1} \approx \frac{2^n}{n}$

$\in \Theta(2^n, n)$

$i-1 \quad n$

d) $\sum_{i=0}^{n-1} \sum_{j=0}^{i-1} (i+j) = \sum_{i=0}^{n-1} \left[ \sum_{j=0}^{i-1} i + \sum_{j=0}^{i-1} j \right]$

$\sum_{i=0}^{n-1} \left[ i(i-1-0+1) + \frac{(i-1)(i-1+1)}{2} \right]$

$= \sum_{i=0}^{n-1} \left[ i^2 + \frac{i^2-i}{2} \right] = \sum_{i=0}^{n-1} \left[ \frac{3i^2-i}{2} \right]$

$= \sum_{i=0}^{n-1} \left[ \frac{3}{2} i^2 - \frac{i}{2} \right] = \frac{3}{2} \sum_{i=0}^{n-1} i^2 - \frac{1}{2} \sum_{i=0}^{n-1} i$

$= \frac{\cancel{3}}{2} \frac{(n-1)(n-1+1)(2n-2+1)}{\cancel{6}\,2} - \frac{1}{2} \frac{(n-1)(n-1+1)}{2}$

$= \frac{1}{4} \underbrace{(n-1).n(2n-2+1)}_{n^3} - \underbrace{(n-1).n}$

$\Theta(n^3) - \Theta(n^2)$

$= \Theta(n^3)$

**4.** Consider the following algorithm.

**ALGORITHM** *Mystery*$(n)$
//Input: A nonnegative integer $n$
$S \leftarrow 0$
**for** $i \leftarrow 1$ **to** $n$ **do**
$S \leftarrow S + i * i$ — $n$
**return** $S$

a) $\sum_{i=0}^{n} i^2 =$

b) multiplication and summation

c) $C(n) = \sum_{i=1}^{n} 1 = n$

d) $\Theta(n)$

e) $\sum_{i=1}^{n} i^2 = \frac{n(n+1) \cdot (2n+1)}{6}$

$\Theta(1)$

**5.** Consider the following algorithm.

```
ALGORITHM Secret(A[0..n − 1])
    //Input: An array A[0..n − 1] of n real numbers
    minval ← A[0]; maxval ← A[0]
    for i ← 1 to n − 1 do
        if A[i] < minval  ✓
            minval ← A[i]
        if A[i] > maxval  ✓
            maxval ← A[i]
    return maxval − minval
```

a) Compute the range

difference between

b) comparison

c) $\sum_{i=1}^{n-1} 2 = (n-1-1) \cdot 2 = 2n-2$

d) $\Theta(n)$

**6.** Consider the following algorithm.

```
ALGORITHM  Enigma(A[0..n − 1, 0..n − 1])
    //Input: A matrix A[0..n − 1, 0..n − 1] of real numbers
    for i ← 0 to n − 2 do
        for j ← i + 1 to n − 1 do
            if A[i, j] ≠ A[j, i]
                return false
    return true
```

a) if input matrix is symmetric, true otherwise false

b) comparison

c) $C_w(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [n-1-i-1+1]$

$= \sum_{i=0}^{n-2} (n-1-i)$

$= (n-1)[n-2-0+1] - \sum_{i=0}^{n-2} i$

$= (n-1)^2 - \frac{(n-1)(n-2)}{2}$

$= \frac{2(n-1)^2 - (n-1)(n+2)}{2}$

$$= \frac{(n-1)\left[2n-\cancel{2}-n+\cancel{2}\right)}{2}$$

$$= \frac{(n-1).(n)}{2}$$

d) $\underline{C_w}(n) = \frac{n(n-1)}{2} \in \underline{\Theta}(n^2)$

$C(n) \in O(n^2)$

e) No,

**2.4**

1. Solve the following recurrence relations.
   a. $x(n) = x(n-1) + 5$ for $n > 1$, $x(1) = 0$
   b. $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$
   c. $x(n) = x(n-1) + n$ for $n > 0$, $x(0) = 0$
   d. $x(n) = x(n/2) + n$ for $n > 1$, $x(1) = 1$ (solve for $n = 2^k$)
   e. $x(n) = x(n/3) + 1$ for $n > 1$, $x(1) = 1$ (solve for $n = 3^k$)

a) $x(n) = x(n-1) + 5$ , for $n \geq 1$

$x(1) = 0$

$x(n-1) = x(n-2) + 5$

$x(n) = x(n-1) + 5$

$= [x(n-2)+5] + 5 = x(n-2) + 2.5$

$= [x(n-3)+5] + 2.5 = x(n-3) + 3.5$

$= \dots$

$= [x(n-i)+5] + (i-1).5 = x(n-i) + i.5$

$= \dots$

$= [x(n-n-1)+5] + (n-1).5$

$= \underbrace{x(1)}_{0} + (n-1).5 \qquad = 5(n-1)$

$x(n) = 5(n-1)$

---

b) $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$

$x(n) = 3x(n-1)$

$= 3(3x(n-2)) = 3^2 x(n-2)$

$= 3^2(3x(n-3)) = 3^3 x(n-3)$

$-$

$$= 3^{i} x(n-i)$$

$$= 3^{n-1} x(\cancel{n}-\cancel{(n}-1)) = 3^{n-1} \underbrace{x(1)}_{4}$$

$$= \underline{4 \cdot 3^{n-1}}$$

c) $x(n) = x(n-1) + n \quad \text{for } n > 0$

$x(0) = 0$

$$x(n) = [x(n-1-1) + n-1] + n = x(n-2) + (n-1) + n$$

$$= [x(n-3) + n-2] + (n-1) + n = x(n-3) + (n-2) + (n-1) + n$$

$$= x[n-n] + (\cancel{n}-\cancel{n}+1) + \dots + n$$

$$= \underbrace{x[0]}_{0} + \underbrace{1 + 2 \dots + n}_{\frac{n \cdot (n+1)}{2}}$$

$$x(n) = \frac{n^2+n}{2}$$

d) $x(n) = x(n/2) + n$ for $n > 1$

$x(1) = 1$

$n = 2^k$

$$x(2^k) = x(2^{k-1}) + 2^k$$
$$= [x(2^{k-2}) + 2^{k-1}] + 2^k$$
$$= x(2^{k-2}) + 2^{k-1} + 2^k$$
$$= [x(2^{k-i-1}) + 2^{k-i}] + 2^{k-i+1} + \dots + 2^k$$
$$= x(2^{k-k}) + 2^{(k-k+1)} + \dots + 2^k$$
$$= \underbrace{x(2^0)}_{1} + 2^1 + 2^2 + \dots + 2^k$$

1

$= 1 + 2^1 + 2^2 + \dots + 2^k$

$= \frac{2^{k+1} - 1}{2 - 1} = 2^{k+1} - 1$

$n$

$x(n) = n \cdot 2 - 1 = \underline{2n - 1}$

e) $x(n) = x(n/3) + 1$ for $n > 1$

$x(1) = 1$

$n = 3^k \longrightarrow$

$x(3^k) = x(3^{k-1}) + 1$

$= [x(3^{k-1-1}) + 1] + 1$

$= x(3^{k-2}) + 2$

$= x(3^{k-i}) + i$

$= x(3^{k-k}) + k$

$= \underbrace{x(3^0)} + k$

$n = 3^k$

$= x(1) + k$ $k = \log_3 n$

$= 1 + \log_3 n$

$x(n) = 1 + \log_3 n$

3. Consider the following recursive algorithm for computing the sum of the first $n$ cubes: $S(n) = 1^3 + 2^3 + \cdots + n^3$.

**ALGORITHM** $S(n)$
//Input: A positive integer $n$
//Output: The sum of the first $n$ cubes
**if** $n = 1$ **return** 1
**else return** $S(n-1) + n * n * n$

**a.** Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.

**b.** How does this algorithm compare with the straightforward nonrecursive algorithm for computing this sum?

a) Basic opt: multiplication $M(n)$

$M(n) = M(n-1) + 2$, $M(1) = 0$

$= [M(n-1-1) + 2] + 2$

$= M(n-2) + 2 + 2$

$= M(n-\underline{\underline{3}}) + \underline{2+2+2}$

$= \dots$

$= M(n-n+1) + \underbrace{2+2+\dots+2}_{n-1}$

$= \underbrace{M(1)}_{0} + 2(n-1)$

$= \underline{2(n-1)}$

b) NonRecS(n)

$s \leftarrow 1$

for $i = 2$ to $n$ do

$s \leftarrow s + \underline{i}\cancel{\times}i\underline{\times i}$

return S

$$\sum_{i=2}^{n} 2 = \underline{2(n-1)}$$

**4.** Consider the following recursive algorithm.

**ALGORITHM** $Q(n)$

//Input: A positive integer $n$
**if** $n = 1$ **return** 1
**else return** $Q(n-1) + 2 * n - 1$

**a.** Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

**b.** Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

**c.** Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it.

a) $Q(n) = Q(n-1) + 2n - 1$

$Q(2) = Q(1) + 2 \cdot 2 - 1 = 4$

$Q(3) = Q(2) + 3 \cdot 2 - 1 = 9$

$Q(4) = Q(3) + 2 \cdot 4 - 1 = 16$

$Q(n) = n^2$

b) $M(n) = M(n-1) + 1, \quad M(1) = 0$

$= [M(n-1-1) + 1] + 1 = M(n-2) + 2$

$= M(n-3) + 3$

$$= M(n-n+1) + n-1 =$$

$$= \underbrace{M(1)}_{0} + n-1 = n-1$$

c) $c(n) = c(n-1) + 3, \; c(1) = 0$

$$= [c(n-1-1)+3]+3$$

$$= c(n-2) + 3+3$$

$$= c(n-i) + i \cdot 3$$

$$= c(\cancel{n}-\cancel{n}+1) + 3(n-1)$$

$$= c(1) + 3(n-1)$$

$$= 0 + 3(n-1) = 3(n-1)$$

$$c(n) = c(n-1) + 2$$

$$= 2(n-1)$$