This problem set has 11 questions, for a total of 100 points. Answer the questions below and mark your answers in the spaces provided. If the question asks for showing your work, you must provide details on how your answer was calculated.

Your Name: _____

1. For each of the following, give an exact formula $T(n)$ for the number of times the line `// op` is run. Show your work and justify your answer. Assume i increments by 1 at each iteration unless otherwise specified.

   (a) [5 points]

```
for (int i = 0 ; i < 4*n ; i ++) {
    // op
}
```

If n= 3, i goes from 0 until 12 but since 12 is not less than 12 the loop only executes from 0-11 which is 12 times

$T(n) = 4n$

(a) _____

   (b) [5 points]

```
for (int i = 1 ; i <= n*n*n ; i++) {
    // op
}
```

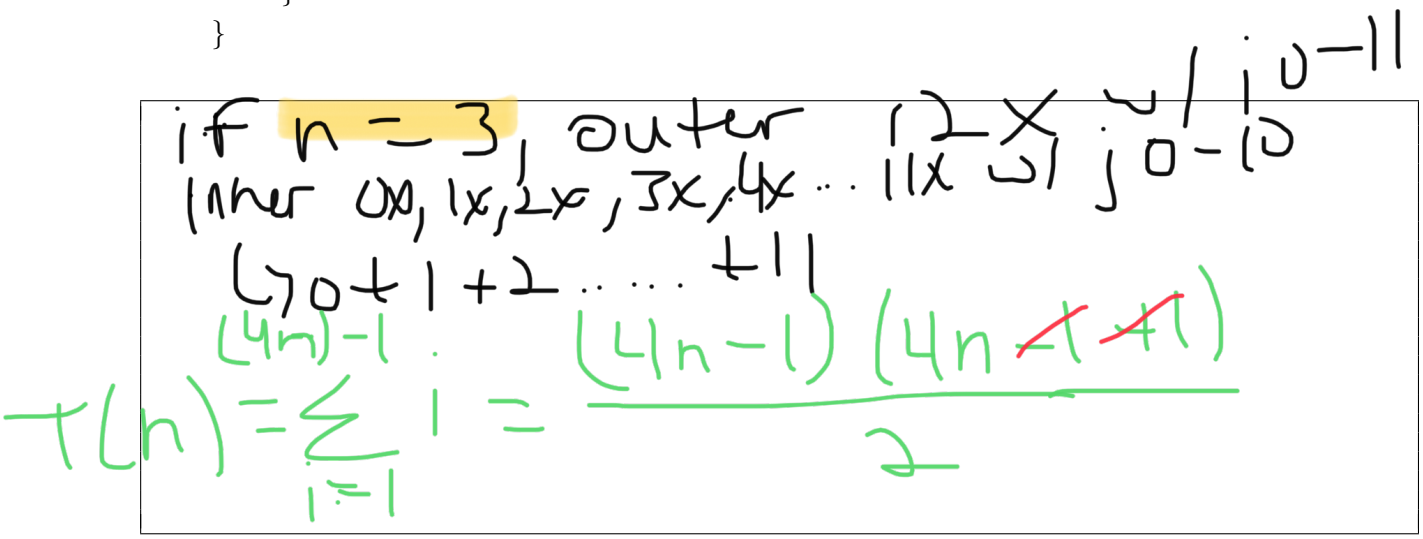
If n= 3, i goes from 1 until 27 and since 27 is less than or equal to 27, the loop executes from 1-27 which is 27 times

$T(n) = n^3$

(b) _____

(c) [5 points]

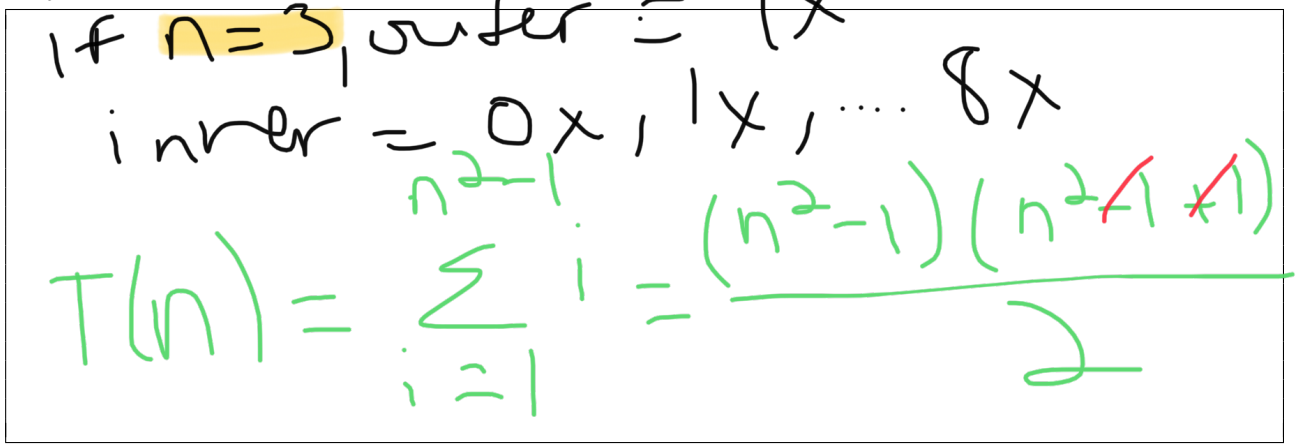
```
for (int i = 0 ; i < 4*n ; i++) {
    for (int j = 0 ; j < i ; j++) {
        // op
    }
}
```



(c) _____

(d) [5 points]

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



(d) _____

(e) [5 points]

```
for (int i = 0 ; i < n ; i++) {
    for (int j = 0 ; j < n ; j++) {
        for (int k = 0 ; k < n ; k++) {
            // op
        }
    }
}
```

n times
n times
n times

if n=3, outer 3x w/ i 0-2
inner 3x w/ j 0-2
inin 3x w/ k 0-2

$T(n) = n^3$

(e) _____

(f) [5 points] Hint: the formula should work with even and odd values of n.

```
for (int i = 0 ; i < n ; i += 2) {
    // op
}
```

if n=1 or 2, i=0
if n=3 or 4, i=0,2
if n=5 or 6, i=0,2,4
if n=7 or 8, i=0,2,4,6

$T(n) = \lceil \frac{n}{2} \rceil$

(f) _____

(g) [5 points]

```
for (int i = 0 ; i < n ; i += 4) {
    // op
}
```

if n = 1 or 2 or 3 or 4, i = 0

if n = 5, 6, 7, 8, i = 0, 4

$T(n) = \lceil \frac{n}{4} \rceil$

(g) _____

(h) [5 points]

```
int m = std::pow(2, n);
for (int i = 1 ; i <= m ; i *= 2) {
    // op
}
```

if n = 3, $m = 2^3 = 8$

i = 1, 2, 4, 8

if n = 7, $m = 2^7 = 128$

i = 1, 2, 4, 8, 16, 32, 64, 128

$T(n) = n + 1$

(h) _____

(i) [5 points] Hint: Assume n is a power of 2

```
for (int i = n ; i > 1 ; i /= 2) {
    // op
}
```

if n = 2, i = 2

if n = 16, i = 16, 8, 4, 2

$T(n) = \log_2(n)$

(i) _____

2. [5 points] Rewrite the following expression into its closed form (i.e. without the sigma): $\sum_{i=1}^{n}(3+i)$. Show your work.

$$\sum_{i=1}^{n}(3+1) = \sum_{i=1}^{n} 3 + \sum_{i=1}^{n} i = 3n + \frac{n(n+1)}{2}$$

A. $3 + \frac{n*(n+1)}{2}$ B. $3 - \frac{n*(n-1)}{2}$ C. $3n - \frac{n*(n+1)}{2}$ D. $3n + \frac{n*(n-1)}{2}$ E. $3n + \frac{n*(n+1)}{2}$

2. _____

3. [5 points] Rank the following functions by their asymptotic growth rate in ascending order.

2 3 1 6 5 4

$\log\log n$ $2^{\log_2 n}$ 2^{100} 4^n $n^2 \log n$ $4^{\log_2 n}$

$b^{\log_b x} = x$

$2^{\log_2 n} = n$

$4^{\log_2 n} = \left(2^2\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^2 = n^2$

4. [10 points] Mark each of the following as true or false.

Upper lower Avg

T(n)	Big O	T/F	Big Omega	T/F	Big Theta	T/F
$\frac{n^2}{10} + 10n \log n$	$O(n \log n)$	F	$\Omega(n \log n)$	T	$\Theta(n \log n)$	F
$2n^2 + n \log n$	$O(n^2)$	T	$\Omega(n)$	T	$\Theta(\log n)$	F
$\frac{n}{2} \log n + 4n$	$O(2^n)$	T	$\Omega(n \log n)$	T	$\Theta(n \log n)$	T
$10\sqrt{n} + 2 \log n$	$O(\log n)$	F	$\Omega(n)$	F	$\Theta(\log n)$	F
$3\sqrt{n} + 10 \log n$	$O(\sqrt{n})$	T	$\Omega(1)$	T	$\Theta(\sqrt{n})$	T

5. [10 points] Complete the following table using Big Θ notation with respect to the number of comparisons.

Ω Θ O

Algorithm	Best Case	Average Case	Worst Case
Selection Sort	n^2	n^2	n^2
Insertion Sort	n	n^2	n^2
Maximum of an Unsorted Array	n	n	n
Median of a Sorted Array	1	1	1
Mode of a Sorted Array	n	n	n

Correct no matter what

hard to do w/o actual paper so just term is fine

6. [5 points] Consider the implementation of *insertion-sort* below.

```
void insertion_sort(int *A, int n) {
    for (int i = 0 ; i < n ; i++) {
        print(A, n);
        for (int j = i ; j > 0 ; j --) {
            if (A[j] < A[j-1]) {
                swap(A, j, j-1);
            } else {
                break;
            }
        }
    }
}
```

loop 1

loop 2

print happens before ith elem that is swapped

Given the array `A` with elements $[2, 44, 21, 9, 1]$ and assuming that `print` sends the current values of `A` to the standard ouput. Show what is printed at every iteration of the outer loop.

i = 0	2	44	21	9	1
i = 1	2	44	21	9	1
i = 2	2	44	21	9	1
i = 3	2	21	44	9	1
i = 4	2	9	21	44	1

loop 1 goes thru every elem while loop 2 goes backward from elem from loop 1 & every time elem in loop 2 is smaller than its prev it sways it else it breaks out of loop 2

7. [5 points] Consider the implementation of *selection-sort* below.

```
void selection_sort(int *A, int n) {
    int min_idx;
    for (int i = 0 ; i < n ; i ++) {
        print(A, n);
        min_idx = i;
        for (int j = i+1 ; j < n ; j ++) {
            if (A[j] < A[min_idx]) {
                min_idx = j;
            }
        }
        swap(A, i, min_idx);
    }
}
```

loop 1

loop 2

print happens before the elem is swapped

Given the array `A` with elements $[2, 44, 21, 9, 1]$ and assuming that `print` sends the current values of `A` to the standard ouput. Show what is printed at every iteration of the outer loop.

i = 0	2	44	21	9	1
i = 1	1	44	21	9	2
i = 2	1	2	21	9	44
i = 3	1	2	9	21	44
i = 4	1	2	9	21	44

loop 1 goes thru each elem while loop 2 finds smallest elem in rest of list to then swap w/ index from loop 1

8. An inversion is any pair of two elements that are out of order. How many inversions are present in each of the following arrays?

(a) [1 point] $[1, 5, 4, 3, 3, 7]$

(a) 5

(b) [1 point] $[5, 4, 3, 2, 1]$

(b) 10

(c) [1 point] $[1, 2, 3, 4, 5]$

(c) 0

(d) [1 point] $[5, 1, 3, 2, 4]$

(d) 5

(e) [1 point] $[6, 9, 1, 4, 10]$

(e) 4

9. Consider the following functions.

```
int foo(int x, int *y) {
    x = x + 10;
    *y = x * 2;
    return x;
}
int *bar(int x) {
    int y = 50 + x;
    return &y;
}
```

For each of the questions below, what are the values of `x` and `y` after running the provided line of code. If you think the code may trigger an error at any point indicate the reason. Do not use a computer for solving this question.

(a) [2 points] `int x = 2, y = 3; x = foo(x, &y);`

(a) x = 12, y = 24

(b) [2 points] `int x = 10, y = 20; x = foo(x, &y);`

(b) x = 20, y = 40

(c) [2 points] `int x = 0, y = 0; x = foo(x, &y);`

x = 10
y = 20

(c) _____

(d) [2 points] `int x = 1, y = 3; int *z = bar(y); x = *z;`

error

(d) _____

(e) [2 points] `int x = 1, y = 0; int *z = bar(y); x = *z;`

error

(e) _____

Optional Questions

The following questions are **optional**, and will not be graded. They are simple problems that should serve as decent practice for solving problems on paper.

10. Write an algorithm in $\Theta(n)$ time that calculates the missing number in an array A of integers. The length of the array is $n-1$ and every element $A[i]$ in the array is such that $1 \leq A[i] \leq n$. For example, given $A = [3, 2, 1, 5]$ the output sould be 4.

11. Write an algorithm that removes all duplicate integers from an input array A. The length of A is n and every element $A[i]$ is such that $1 \leq A[i] \leq 100$. For example, given $A = [12, 2, 2, 3, 4, 2, 5]$, the algorithm should return $[12, 2, 3, 4, 5]$. Can you make your algorithm run in $\Theta(n)$ time?

qc
return to x
new x = 0 + 10 = 10
same y = 10 * 2 = 20

qd & e
can't return ref to local var b/c it no longer exists once func call is over

10
Done in Lab 5
↳ find sum of 1-n then subtract each elem in arr from it's missing num

11
O(n²) ⇒ nested loops to find it elem exists before insert
O(n log n) ⇒ sort then iter
O(n) ⇒ hash (w/ all)