CS427 Fall Assignment #1

Contents: Chapter 1-2.2

Textbook, Pg.8 Ex.10 (a), Pg.18 Ex.9, Pg.59 Ex.3

Rules:

* Each pseudocode must specify its input and output.
* Necessary comments should be included.
* When analyzing complexity, specific reasons must be provided.
* For proof questions, a specific derivation process is required. Please refer to the example in the slide.

1. Euclid’s algorithm, as presented in Euclid’s treatise, uses subtractions rather than integer divisions. Write pseudocode for this version of Euclid’s algorithm. (10 pts)

// Time Complexity O(n) – While loop can run up to n iterations if a and b are large and nieghboring.

// Space Complexity O(1) – no additional frames or functions called

EuclidsAlg(int a, int b) {

// Input: two positive integers

// O(n)

While(a != b) {

// The GCD of two numbers does not change if you replace the larger number with the difference of the two

//O(1)

If(a > b) {

a = a – b;

} else {

b = b – a;

}

}

// Output: Returns GCD (a) when a and b are equal

Return a;

}

2. Consider the following algorithm for finding the distance between the two closest elements in an array of numbers. (10 pts)

**ALGORITHM *MinDistance*(A[0..n − 1])**

**//Input: Array A[0..n − 1] of numbers**

**//Output: Minimum distance between two of its elements**

***dmin* ← ∞  
for i ← 0 to n − 1 do**

**for j ←0 to n−1 do  
 if ij and |A[i]−A[j]| < *dmin***

***dmin* ← |A[i]−A[j]|**

**return *dmin***

Make as many improvements as you can in this algorithmic solution to the problem. If you need to, you may change the algorithm altogether; if not, improve the implementation given.

// Time Complexity O(n log n) – Using Arrays.Sort() using dual pivot quicksort with O(n) time to compare each element with pivots and O(log n) to divide the array into constant-fraction pieces

// Space Complexity: O(1) - with primitive ints in Arrays.Sort() create negligible space overhead

**ALGORITHM *MinDistance*(A[0..n − 1])**

**//Input: Array A[0..n − 1] of numbers**

**//Output: Minimum distance between two of its elements**

Int dmin = +**∞**

// Exit if array is less than 2 numbers

//O(1)

If(A.length < 2) {

return dmin;

}

//Sort numbers O(n log n)

Arrays.Sort(A);

//O(n)

For(int i = 1; i < A.length; i++) {  
 // If neighboring numbers are the same – we return 0

If (A[i] == A[i-1]) {

Return 0;

}

if(A[i] – A[i-1] < dmin) {

dmin = A[i] – A[i-1];

}

}

return dmin;

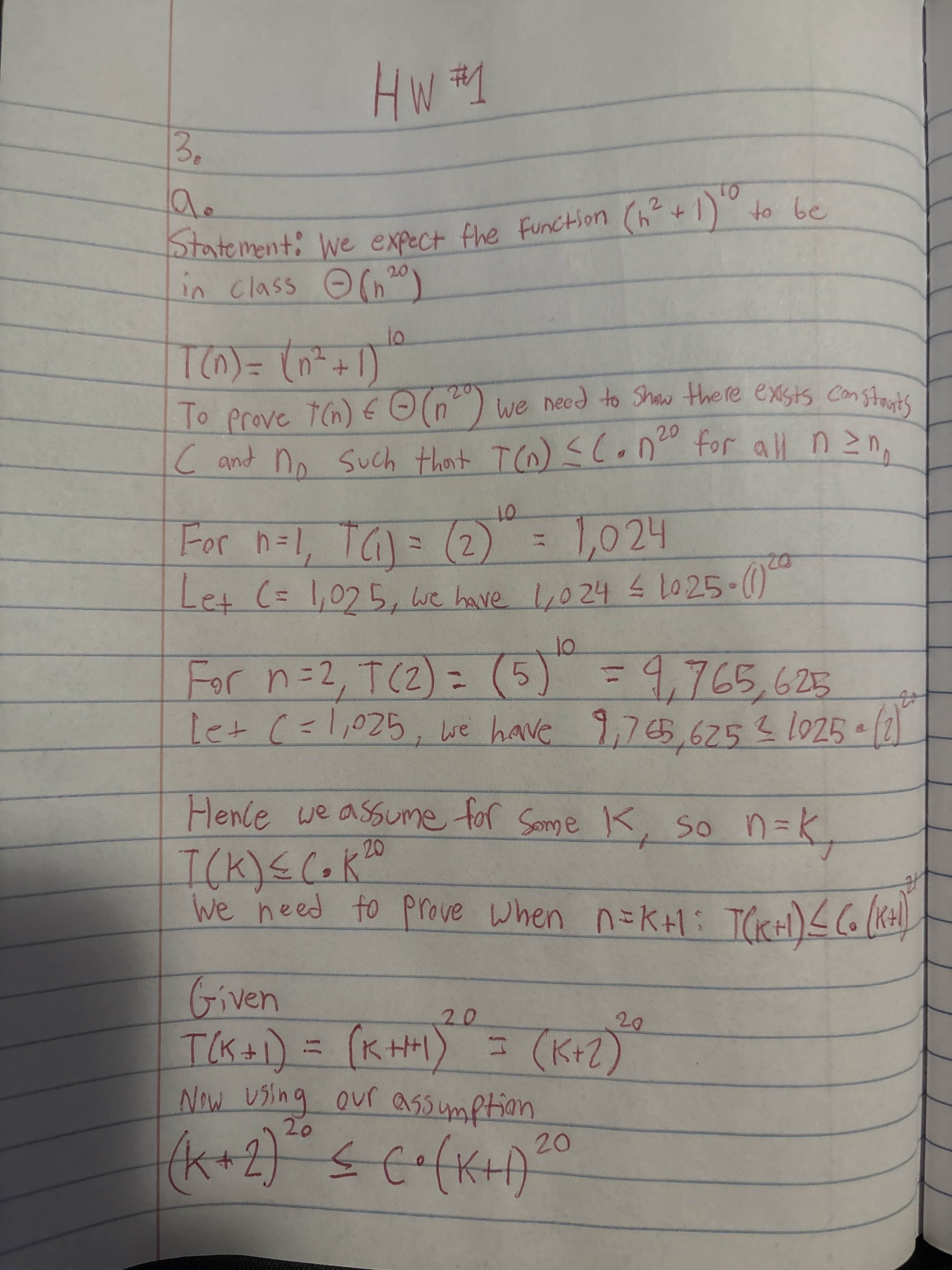
}

3. For each of the following functions, indicate the class Θ(g(n)) the function belongs to. (Use the simplest g(n) possible in your answers.) Prove your assertions. (75 pts)

Note: Please follow the assignment format of the slides.

For each question, you need to provide **Statement**, the **Prove** of mathematical derivation, and **Conclusion**.

**a.** (n2 +1)10



A notebook with writing on it

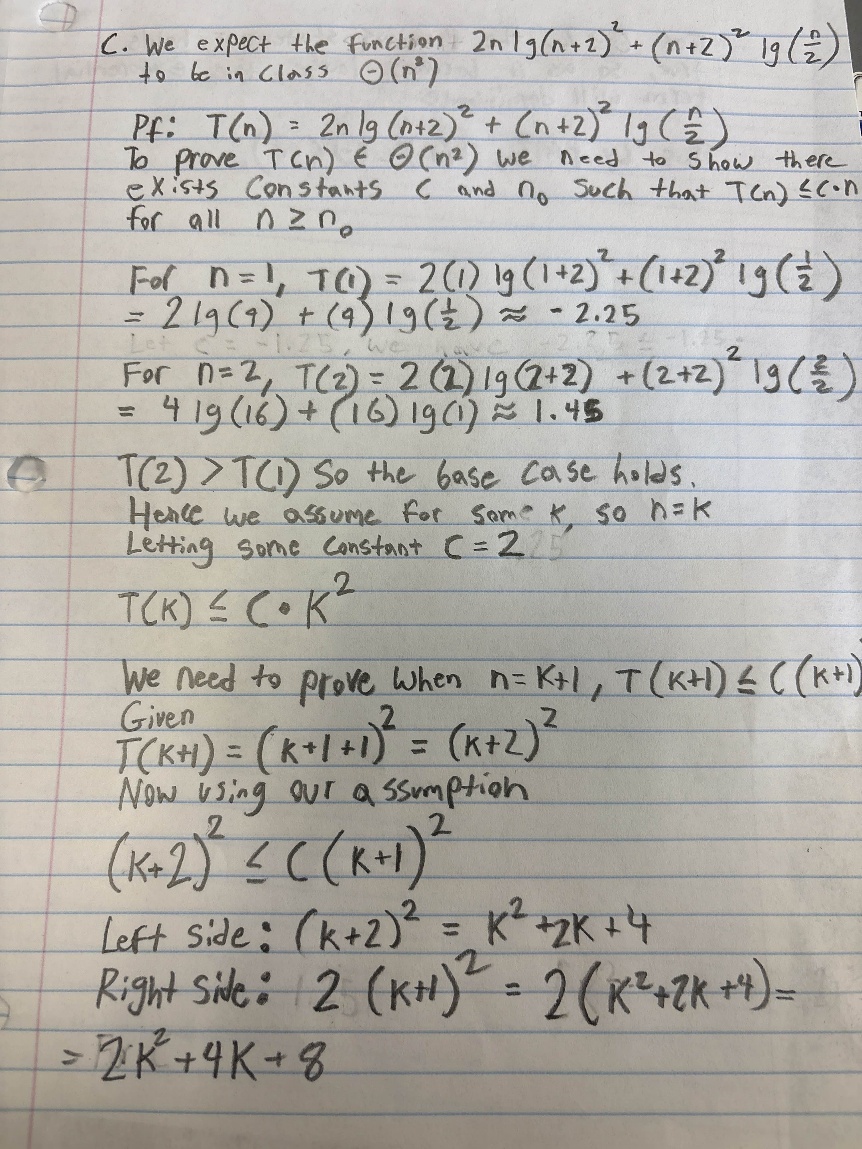
AI-generated content may be incorrect.

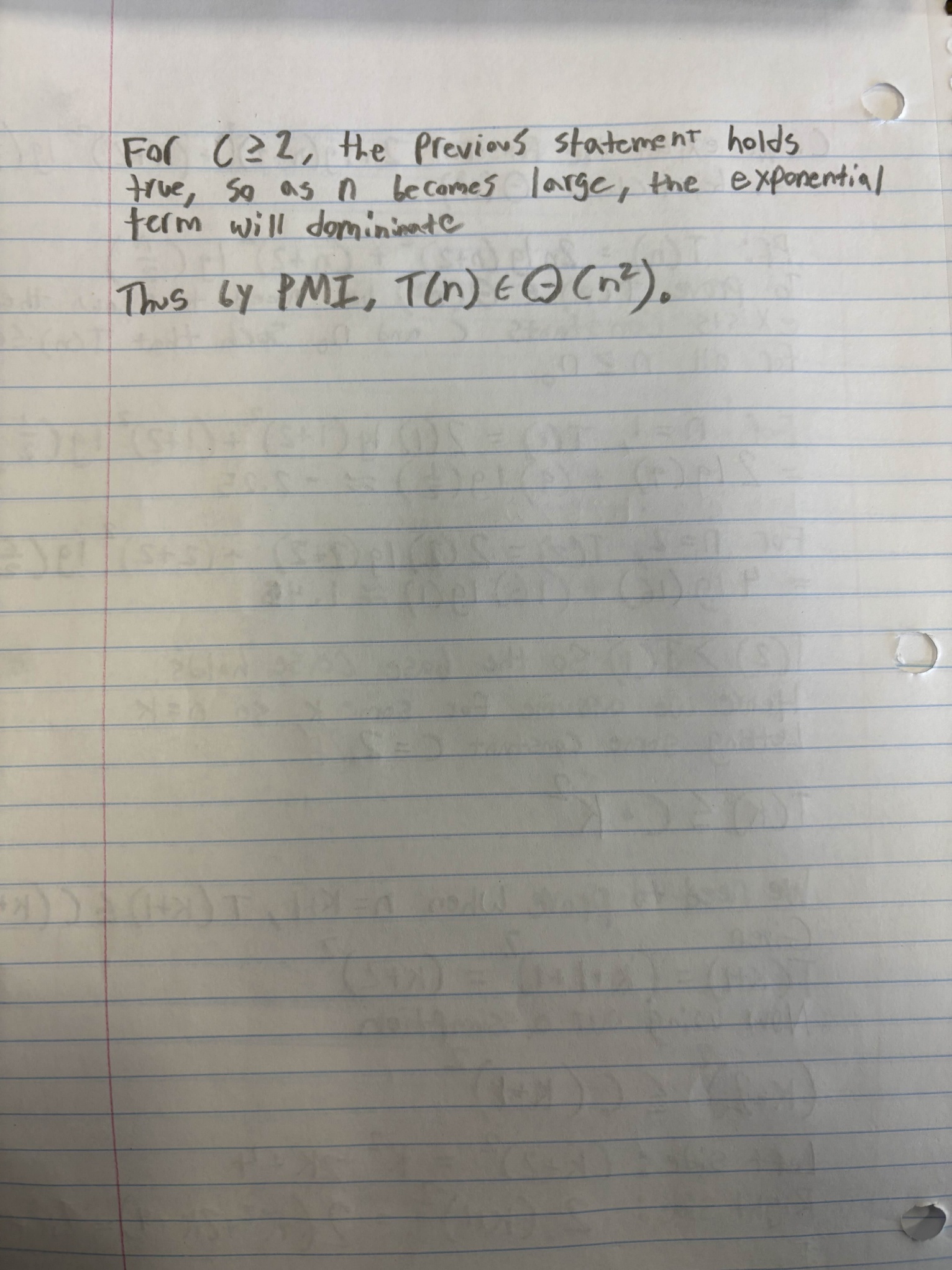
**b.**

**A notebook with writing on it

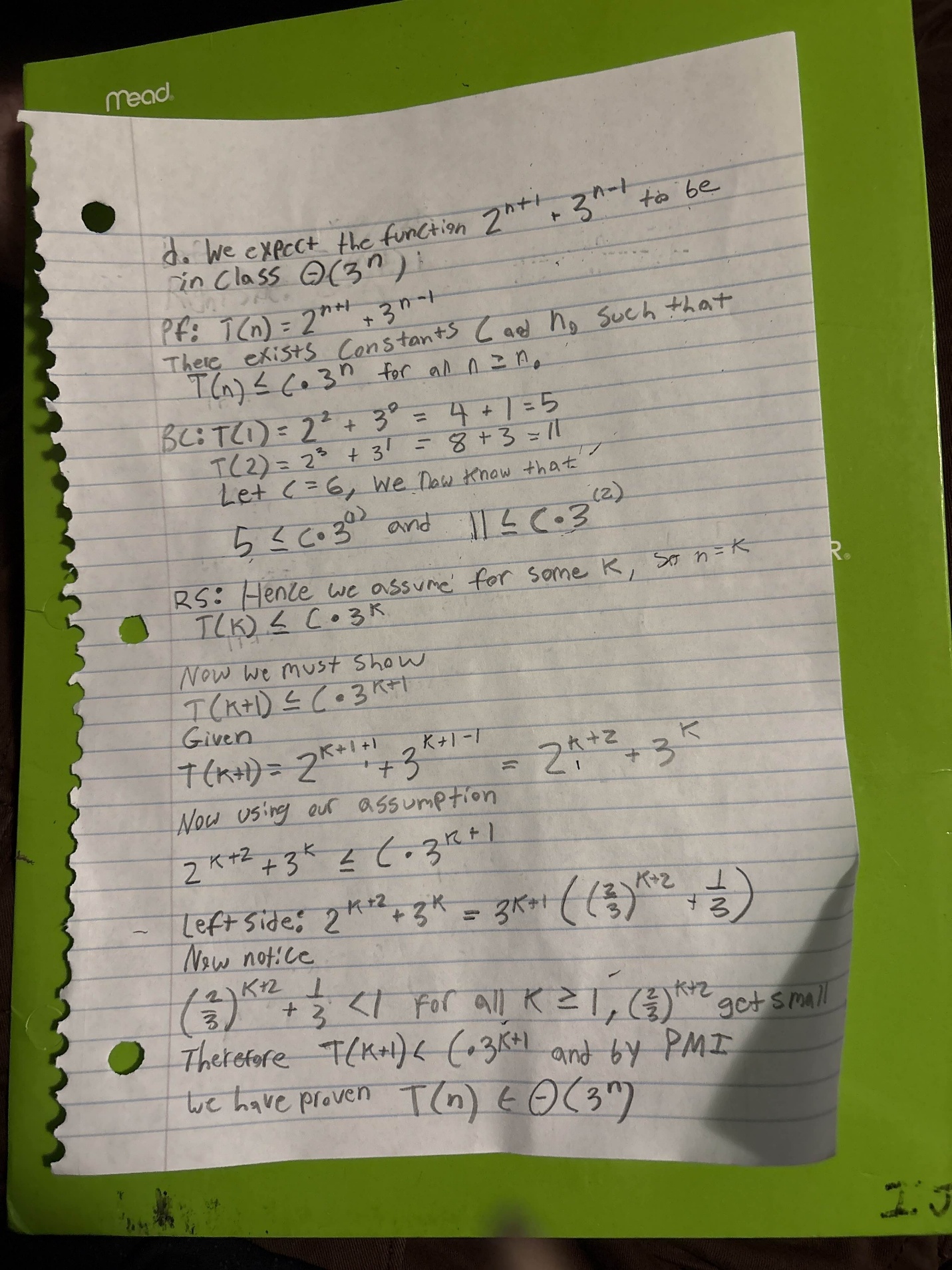
AI-generated content may be incorrect.**

**c.** 2n lg(n+2)2 +(n+2)2 lg





**d.** 2n+1 +3n−1



**e.** ⌊log2 n⌋

