Name: ________________________________

Here are some basic rules for calculating the Big O for some $T(n)$ or an algorithm.

1. Only the highest degree of n matters. For example

   $$T(n) = n^3 + 5n^2 + 10^7 \to O(n^3)$$

   since once n becomes super-massively huge, the other terms just stop mattering.

2. Constant factors don't matter. $T(n) = 500n$ and $T(n) = 0.005n$ both $O(n)$. Again, as n becomes bigger, these constants stop mattering; what matters is the rate of growth. Example:

   $$T(n) = 50n^3 - 2n^2 + 400 \to O(n^3)$$

3. Counting the number of nested loops usually works.

You can turn in this assignment physically to a TA or me. You can also scan and upload your answers.

For each of the following $T(n)$, write the corresponding Big O time complexity. Some series may require research.

1. (2 points) $T(n) = n^2 + 3n + 2$

   1. ______________

2. (2 points) $T(n) = (n^2 + n)(n^2 + \frac{\pi}{2})$

   2. ______________

3. (2 points) $T(n) = 1 + 2 + 3 + \ldots + n - 1 + n$

   3. ______________

4. (2 points) $T(n) = 1^2 + 2^2 + 3^2 + \ldots + (n-1)^2 + n^2$

   4. ______________

5. (2 points) $T(n) = 10$

   5. ______________

6. (2 points) $T(n) = 10^{100}$

   6. ______________

7. (2 points) $T(n) = n + \log n$

   7. ______________

8. (2 points) $T(n) = 12\log(n) + \frac{n}{2} - 400$

   8. ______________

9. (2 points) $T(n) = (n+1) \cdot \log(n) - n$

   9. ______________

10. (2 points) $T(n) = \frac{n^4+3n^2+2n}{n}$

   10. ______________

______________ 20 points

11. (4 points) What is the time complexity to get an item from a specific index in an ArrayList?

12. (3 points) What is the time complexity remove an item in the middle of an ArrayList?

13. (3 points) Why?

14. (3 points) What is the **average** time complexity to add an item to the end of an ArrayList?

15. (3 points) What is the **worst case** time complexity to add an item to the end of an ArrayList? What if you have to or don't have to reallocate?

16. (4 points) Taking this all into account, what situations would an ArrayList be the appropriate data structure for storing your data?

_______________ 20 points

```
public static int[] allEvensUnder(int limit){
        if (limit <= 0){
                return new int[0];
        }
        if (limit < 2){
                return new int[1];
        }
        int[] vals =  new int[(limit+1)/2];
        for(int i  = 0;  i <(limit+ 1)/2  ; i++ ) {
                vals[i] = i*2;
        }
        return vals;
}
```

17. (5 points) What is the **time** complexity of the above algorithm?

18. (5 points) What is the **space** complexity of the above algorithm? In other words, how much space is used up as a function if the input size? Think about it, we didn't cover this in the videos.

```
/*
*  https://rosettacode.org/wiki/Sorting_algorithms/Insertion_sort#Java
*/
public static void insertSort(int[] A){
        for(int i = 1; i < A.length; i++){
                int value = A[i];
                int j = i - 1;
                while(j >= 0 && A[j] > value){
                        A[j + 1] = A[j];
                        j = j - 1;
                }
                A[j + 1] = value;
        }
}
```

19. (10 points) What is the time complexity of the above algorithm?

`bogosort` attempts to sort a list by shuffling the items in the list. If the list is unsorted after shuffling, we continue shuffling the list and checking until it is finally sorted.

20. (5 points) What is the worst case run time for `bogosort`?

21. (5 points) Why?

22. (5 points) What is the average case run time for `bogosort` (Hint: think about a deck of cards)?

23. (5 points) Why?

24. (20 points) For each of the methods you wrote in Lab 2, figure out the time complexity of the method you wrote. To turn in this portion, attach a printout of the code and specify the time complexity of each.

_______________ 20 points