Consider the following algorithm for sorting a list of numbers.

Procedure BubbleSort(A) {
    n = A.length;
    for i = 1 to n - 1;
        for j = i + 1 to n;
            if A[j] < A(i) {
                t = A[j];
                A[j] = A[i];
                A[i] = t;
            }
    }

A problem of size $n$ is a list of $n$ integers, which without loss of generality, can be taken to be the integers 1 through $n$. An instance of a size-$n$ problem is a particular permutation of the list.

1. For which instance is the instruction count for the algorithm the smallest? Call this instance the best case.

2. For which instance is the instruction count the greatest? Call this instance the worst case.

3. For the best case, identified above, let $T_1(n)$ be the instruction count. Find an exact expression for $T_1(n)$, free of summations, and specify its asymptotic growth rate.

4. For the worst case, identified in item 2 above, let $T_2(n)$ be the instruction count. Find an exact expression for $T_2(n)$, free of summations, and specify its asymptotic growth rate.

5. Assuming that all $n!$ permutations of the input are equally probable, let the random variable $T_3(n)$ be the instruction count on a random run of the algorithm. In this case, it is more difficult than the in-class example to find an exact expression for $E[T_3]$. However, you should be able to derive the asymptotic growth rate of $E[T_3]$ by reflecting on the relationships between an average score and the extreme (worst and best) scores.
Solutions:

1. All statements, except those in the scope of the if-test, are executed the same number of times regardless of the input permutation. The best (smallest) instruction count then occurs if the if-test always fails. That is, for each $1 \leq i \leq n - 1$, we find $A[j] \geq A[i]$ for all $j > i$. That is, $A[2]$ through $A[n]$ are all greater than $A[1]$. Also, $A[3]$ through $A[n]$ are all greater than $A[2]$, and so forth. We conclude that the smallest instruction count (best case) occurs when the input is in increasing order.


3. Let $t_i = n - (i + 1) + 1 + 1 = n - i + 1$ be the number of times the for-j loop test executes during the $i^{th}$ repetition of the i-loop. Since the best case performs no executions of the if-statement body, we have

```plaintext
Procedure BubbleSort(A) {
    n = A.length;
    for i = 1 to n - 1;
        for j = i + 1 to n;
                t = A[j];
                A[j] = A[i];
                A[i] = t;
            }
    T_1(n) = 1 + n + \sum_{i=1}^{n-1} t_i + \sum_{i=1}^{n-1} (t_i - 1)
            = 1 + n - (n - 1) + 2 \sum_{i=1}^{n-1} t_i
            = 2 + 2 \sum_{i=1}^{n-1} (n - i + 1)
            = 2 + 2(n - 1) + 2 \sum_{i=1}^{n-1} (n - i) = 2n + 2[(n - 1) + (n - 2) + \ldots + 1]
            = 2n + 2[1 + 2 + \ldots + (n - 1)] = 2n + 2 \sum_{i=1}^{n-1} i
            = 2n + 2 \cdot \frac{n(n - 1)}{2}
            = n^2 + n.
}
```

Noting $n^2 \leq n^2 + n \leq 2n^2$, for all $n \geq 1$, we conclude $T_1 = \Theta(n^2)$.

4. The worst case performs the if-statement body every time the if-test itself executes. In the expression for $T_1$, the if-statement test gives rise to the term $\sum_{i=1}^{n-1} (t_i - 1)$, which counts the total executions of the if-test. In the worst case, each such test results in three more statement executions. So, an expression for $T_2$ contains all the terms of $T_1$, with the if-statement count multiplied by 4. That is,

```plaintext
T_2(n) = 1 + n + \sum_{i=1}^{n-1} t_i + 4 \sum_{i=1}^{n-1} (t_i - 1) = 1 + n - 4(n - 1) + 5 \sum_{i=1}^{n-1} t_i = -3n + 5 + 5 \sum_{i=1}^{n-1} (n - i + 1)
        = -3n + 5 + 5(n - 1) + 5 \sum_{i=1}^{n-1} (n - i) = 2n + 5 \sum_{i=1}^{n-1} i = 2n + 5 \cdot \frac{n(n - 1)}{2}
        = \frac{5}{2}n^2 - \frac{1}{2}n.
```

We have $n^2/2 \geq n/2$ for all $n$, so

$$2n^2 = \frac{4}{2}n^2 = \frac{5}{2}n^2 - \frac{1}{2}n^2 \leq \frac{5}{2}n^2 - \frac{1}{2}n \leq \frac{5}{2}n^2.$$

We conclude that $T_2 = \Theta(n^2)$. 

2
5. On each of the $n!$ permutations of a sorted input, the algorithm instruction count will be greater than or equal to the best case. So,

$$E[T_3] = \frac{1}{n!} \sum_{i=1}^{n!} \tau_i(n) \geq \frac{1}{n!} \sum_{i=1}^{n!} T_1(n) = T_1(n) \geq n^2,$$

where $\tau_i(n)$ is the instruction count on the $i^{th}$ permutation of the sorted input. Also,

$$E[T_3] = \frac{1}{n!} \sum_{i=1}^{n!} \tau_i(n) \leq \frac{1}{n!} \sum_{i=1}^{n!} T_2(n) = T_2(n) \leq \frac{5}{2} n^2.$$

Therefore, $n^2 \leq E[T_3] \leq (5/2)n^2$, and $E[T_3] = \Theta(n^2)$ as well.