Exercises for the Midterm

Question 1

Write an algorithm in pseudo-code that accepts an integer n and an unordered array of n integers. The algorithm should calculate the average of the numbers and then tell how many of the elements are above the average. Exactly how many additions are done in the algorithm and exactly how many comparisons are done.

Question 2

The idea of binary search can be generalized into k-ary search by dividing an n-element sorted array into n/k sub-arrays. Devise a tenary search algorithm that first tests the element at position n/3, and then checks the element at 2n/3. This algorithm either discovers the target element or reduces the array to one-third of the original size.

Question 3

Let $f(n) = 56 \times n^2$ and $g(n) = 54 \times n^3$. Which of the following statements are true? (T or F. No explanation needed.)

1. $f(n) \in \omega(g(n))$
2. $g(n) \in \omega(f(n))$
3. $f(n) \in \theta(g(n))$
4. $g(n) \in \theta(f(n))$
5. $f(n) \in o(g(n))$
6. $f(n) \in O(g(n))$
7. $f(n) \in \Omega(g(n))$
8. $g(n) \in \Omega(f(n))$

Question 4

Given a recurrence relation $T(n) = 3T(n/2) + n^2$, $T(1) = 1$.

1. Draw the first four levels of the recurrence tree.
2. Solve the recurrence to find a formula for $T(n)$.

Question 5

The following procedure uses A[r] as a pivot to partition the array A[p..r]:

```
PARTITION(A,p,r)
    pivot = A[r]
    i = p-1
    for j = p to r-1
        if A[j] <= pivot
            i = i+1
            swap A[i] and A[j]
    swap A[i+1] and A[r]
    return i+1
```

5.1 What is the loop invariant of the for loop.

5.2 What value does PARTITION return when all elements in A[p..r] are the same?

5.3 Modify the procedure so that it returns (p+r) div 2 when all elements in A[p..r] are the same.

Question 6

The following table compares several sorting algorithms, where the column Best gives the best-case running time, Worst gives the worst-case running time, and In-place indicates if the algorithm is in-place. Complete the table.

Name	Best	Worst	In-place
selection sort	$\Theta(n^2)$	$\Theta(n^2)$	yes
insertion sort			
bubble sort			
quick sort			
merge sort			
heap sort			