Problem 1. Use the integral method to get upper and lower bounds for $\sum_{j=0}^{n} j^2$. The two values should have exactly the same higher order term. Show your work.

Problem 2. Consider the following quicksort-like sorting algorithm. Pick two elements of the list. Partition based on both of the elements. So the elements smaller than both are to the left, the elements in between are in the middle, and the elements larger than both are to the right.

1. Give a brief English description / pseudo-code of an algorithm of how you would partition. Try to minimize the number of comparisons.
2. How many comparisons does the partition algorithm use in the worst case? Show your work.
3. How many comparisons does the partition algorithm use on average. Just get the high order term. Justify informally.
4. Assume that the two partition elements always partition exactly into thirds. Write a recurrence for the number of comparisons. Solve this recurrence using constructive induction. Just get the high order term exactly.
5. Assume that the two partition elements always partition so exactly one quarter are to the left, one half in the middle, and one quarter to the right. Write a recurrence for the number of comparisons. Solve this recurrence using constructive induction. Just get the high order term exactly.

Problem 3. Given an array of integers, A, such that, for all $i, 1 \leq i < n$, we have $|A[i] - A[i+1]| \leq 1$. Let $A[1] = x$ amd $A[n] = y$, such that $x < y$. Design an efficient search algorithm to find j such that $A[j] = z$ for a given value $z, x \leq z \leq y$. What is the maximal number of comparisons to z that your algorithm makes?