1. Do not use integrals for this problem. Do not worry about floors and ceilings (so you may assume that n is "nice"). Ignore second order terms. Consider

$$\sum_{k=1}^{n} k^{1.5} .$$

   (a) Split the sum into two equal-sized regions to obtain an upper bound for its value.

   (b) Split the sum into two equal-sized regions to obtain a lower bound for its value (as done in class).

2. (a) Use the integral method to obtain upper and lower bound bounds for

$$\sum_{k=1}^{n} k^{1.5} .$$

   (b) How do your bounds compare with those obtained in Problem 1?

3. Consider an array of size eight with the numbers $30, 40, 80, 70, 10, 20, 50, 60$. Assume you execute quicksort using the version of partition from class and from CLRS. Note that in this algorithm an element might exchange with itself (which counts as one exchange).

   (a) Show the array after the first partition. How many comparisons are used? How many exchanges?

   (b) Show the left side (of the original pivot) after the next partition. How many comparisons are used? How many exchanges?

   (c) Show the right side (of the original pivot) after the next partition on that side. How many comparisons are used? How many exchanges?

4. We are going to derive an upper bound for the average number of exchanges for quicksort. A similar analysis would give a lower bound, giving the high order term exactly.

   (a) Assume that the partition (or pivot) element ends up in position q. How many exchanges does partition do, NOT counting the final exchange where the pivot element is placed in its proper sorted position? Briefly justify. Note that an element can exchange with itself.

   (b) Write a recurrence for the expected number of exchanges (for quicksort), NOT counting the final exchange where pivot element is placed in its proper sorted position.

   (c) Simplify the recurrence as much as reasonably possible (as we did in class for comparisons).

   (d) Guess that the solution is at most $an \ln n$ for some constant a. Use constructive induction to verify the guess and derive the constant a.

   (e) Give an upper bound on how many exchanges involve the pivot element thoughout all of the partitions in the entire quicksort algorithm. Briefly justify.

   (f) Add this value to your answer in Part (d) to get an upper bound on the total number of exchanges.

   (g) Rewrite your solution using log base 2 rather than the natural log, evaluating the constant to three decimal places.

5. **Challenge problem, will not be graded.** Calculate the exact average number of exchanges involving the pivot in quicksort.