1. A program uses one unit of time for the first case, two units of time for the second case, etc. (i units of time for the i-th case). If there are n cases, what is the total time that the program uses?

2a. Define $B_n = nB_{n-1}$ for $n > 1$, $B_1 = 1$. Give a simple closed form for B_n.

2b. Define $A_n = nA_{n-1} + B_{n-1}$ for $n > 1$, $A_1 = 1$, where B_n is defined above. Give as simple an expression as you can for A_n.

3. Find upper and lower bound for

$$\sum_{n \leq i \leq \infty} \frac{x^{2i+1}}{2i+1}$$

This sum does not converge for all x, so also give the range of x your bounds work. An ideal answer to this question will have upper and lower bounds that are close together and that are also simple. In case you need to make a trade-off based on the value of x, we are most interested in the case where $x = 1/3$.

4. One way of doing the project for the course leads to an error term whose absolute value is $B_{2m}/(2mn^{2m})$, where B_i is the i-Bernoulli number.

   4a. When m is fixed, how does this error term vary with n? Is there a best value of n (assuming we are trying to reduce this error term without regard to how much time is needed for the computation)?

   4b. When n is fixed, how does this error term vary with m? Is there a best value of m (assuming we are trying to reduce this error term without regard to how much time is needed for the computation)?

5. An alternate analysis of Algorithm 5.10 (Select) leads to the recurrence

$$C(n) \leq C\left(\frac{n}{c}\right) + C\left(\frac{3c-1}{4c}n - \frac{c+1}{4}\right) + \alpha n.$$

The solution to this recurrence can be bounded by

$$C(n) \leq \beta n.$$

What is the best value you can obtain for β? The best value of β clearly depends on c, α, and the boundary conditions. To simplify the problem, assume that the boundary conditions are favorable, so that they do not limit the performance of the algorithm. Give enough details to show that you have a correct answer.