

For #2 (and other problems involving sums and/or recurrence relations), the intent is to use the sums and recurrence relations tables from class to get the big-Oh. Other strategies, such as using the closed form solution for the sum or applying the master theorem from ADM 5.4, are valid but make sure you know how to use the tables because only those will be provided for reference on exams.

It is legal but not necessary to use the definition of Θ to justify your answers for #2 — you can just drop constant factors and additive terms (and state that as your justification). If you do use the definition of Θ , it should be clear to both you and your reader that the values of c and n_0 you pick are correct.

When comparing the growth rates of two functions as in #3, simplify the functions as much as possible — the goal is to get things as close as possible to one of the known simple functions 1 , $\log n$, n , etc. Factoring out common factors helps e.g. $n \log n$ and $n^{1.5}$ have a factor of n in common ($n \log n$ vs $n \cdot n^{0.5}$) so the comparison is really between $\log n$ and $n^{0.5}$ — now it is possible to appeal to the known simple functions and say that polynomials grow faster than logs.

For #4 and #6, write the sums for the loops and use the sums table to solve them. Assuming that every loop repeats n times overcounts — this may be safe (in the sense that it doesn't change the big-Oh), but it may not be. #4 specifically asked you to write the sums rather than simply rounding up in order to starting building a sense for when rounding up is OK vs when it isn't.

For #5 and #6, make sure your answers come from the code and not from assumptions that a loop always repeats n times or previous knowledge about bubble or comb sort. In #5, for example, the inner `for` loop is a counting loop so it is straightforward to see how many times it repeats, but the outer loop repeats until a condition is true — how many times is that? It is necessary to analyze what is going on in the `for` loop, and to cite that analysis in your answer.