Problem 1

The answer can be got from inspection as in the following two tables.

i	j	Printing #
1	1 to 4	4
2	2 to 7	6
3	3 to 10	8
…	…	…

n	i	Printing #
1	1 to 1	4
2	1 to 2	10
3	1 to 3	18
…	…	…

But you must offer some concrete "evidence" to say the answer is something. Inspection is not good enough for such a kind course. (Algorithm)

The inner loop is only a print statement, so the constant execution time is needed. You can use a "c" or "k" to denote the constant, or just using 1 (unity). Either is O (1).

Take a look at your textbook pp.1058-1060 for some concepts about summations, and don't been fooled by the index term (*k* in the following example) and the summed term (*b* in the following example).

$$\sum_{k=a}^{a+2} b = \underset{k=a}{b} + \underset{k=a+1}{b} + \underset{k=a+1}{b} = 3b$$

In the above example, the exact value of *a* is NOT important, as long as it won't change during summation. The same idea applies to professors' solution. When doing

$$\sum_{j=i}^{3i+1} 1 = \underset{j=i}{1} + \underset{j=i+1}{1} + \ldots + \underset{j=3i+1}{1} = (3i+1) - (i) + 1 = 2i + 2$$

i won't change during the summation, which can be known by inspecting the code. The second equal sign can be got from a simple counting. I (TA) learned it when I was 10 years old.

Problem 2

Read the English again if you failed to catch the meaning of "minimize the maximum sum of a pair". For example, if the input is {1,9.5,3}, one partition is {1,9} and {5,3}, and the maximum sum is *max (1+9, 5+3)* = 10. Other partitions exist also, like {1,5} and {9,3}, whose output is *max (1+5,9+3)*= 12. The question asks, "How to pair to get the minimize *max(...)* ? "Once you grasp the meaning of questions, it's not hard to guess the correct answer.

Problem 3(a)

Sorting, a topic will be covered in the near future, needs some time. In fact, there is a lower bond, $\Omega(n\log n)$, for comparison-based sorting, which is covered on your textbook pp. 165-167. Namely, if you want to do something via sorting, the worse-case running time is at lease $O(n\log n)$.

Problem 4

Don't forget the base case ($n = 1$) in induction. If you know something about strong induction and weak induction, weak one is enough in this question. If not, ignore the previous sentence.