Problem 1

Suppose you want to calculate $a^{16} mod\ p$. The straightforward way is to multiply a, 16 times which requires 16 multiplications. However, the smarter way is to calculate a^2, then calculate $a^4 = a^2 * a^2$ by one multiplication, then $a^8 = a^4 * a^4$, and finally calculate $a^{16} = a^8 * a^8$. This method requires only 4 multiplications. If b is not a proper power of 2 then the same idea with a few changes can be applied. Don't forget to take mod whenever you multiply to avoid overflow.

Problem 2

The idea is to use the merge sort algorithm. Suppose your list is $A = (a_1, a_2, \dots, a_{16})$, and you divide the list into two sub-lists L=$(a_1, \dots, a_8)$ and R=$(a_9, \dots, a_{16})$. You can solve each subproblem recursively, and find the inversion number for both sub-lists. The only problem is to calculate the number of pairs with the desired properties that have one element in the left list and one element in the right list. This can be calculated when you are merging the two sub-lists.

Problem 3

You can check if a configuration with parameter X is feasible by a greedy algorithm. The greedy algorithm starts with the leftmost house and put a router in distance X in its right side. Then, The greedy algorithm removes all the covered houses. It will continue doing this until all the houses are covered.

In order to find the smallest feasible X^*, you can use binary search.