This is an individual/group homework assignment. Groups of **up to 2** students can submit joint solutions. **EXACTLY** one student in the group should submit the solution document, and at the beginning of the document, you should clearly **state the names of all groupmates**. The student who submits the solution on behalf of the group is responsible for sharing the grading feedback with the rest of the group. If a groupmate's name is missing on the submitted file, that student will NOT receive any points for the corresponding assignment.

No credit will be given for handwritten solutions (except for figures or long equations that might be handwritten). You must submit your assignment on Canvas in the related digital repository by the specified deadline.

As it appears in the course syllabus, for the homework assignments, students are encouraged to discuss the problems with others, but you are expected to turn in the results of your own effort (not the results of a friend's efforts in another group or so). Even when not explicitly asked, you are supposed to justify your answers concisely and clearly.

Question 1.

(a) Write a linear time divide and conquer algorithm (i.e., $\theta(n)$) to calculate x^n (x is raised to the power n). Assume x and n are integers >=0. Note: your algorithm here should be of $\theta(n)$.

(b) Analyze the time complexity of your algorithm in the worst-case by first writing its recurrence relation. Show why it is $\theta(n)$.

(c) Can you improve your algorithm to accomplish the end in O(log n) time complexity (we still look for a divide and conquer algorithm). If yes, write the corresponding algorithm, write the recurrence relation for its time complexity and analyze it. If no, justify your answer.

Question 2.

a) Explain the divide and conquer algorithmic technique in general. How do we usually analyze the time complexity of algorithms written using this technique. Consider an example and explain.

b) Quick sort algorithm uses a divide and conquer strategy to sort the given list. Assume the pivot is always defined such that it divides up the list into two subarrays of equal size. Analyze your algorithm by writing the recurrence relation. Apply Master Theorem to bound the relation asymptotically. Show all your work.

Question 3:

(a). Let's consider a long, quiet country road with houses scattered very sparsely along it. (Picture the road as a long line segment with an eastern endpoint and a western endpoint.) Further let's suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal using as few base stations as possible. Prove its correctness and explain its time complexity.

(b). Implement your algorithm in part a in C++/Java. Given a list of n house positions, output the positions of base stations for optimal outcome. Paste your code in the solution file. Comment as needed. Analyze it's time complexity providing enough details and justification. Show all your work.