1. (10 pts) Consider a long quiet country road with houses scattered very sparsely along it. We can picture the road as a line segment, with an eastern endpoint and a western endpoint. We want to place cell phone towers at certain points along the road so that every house is within 2 miles of one of the base stations. The input to the program is the location of the \( n \) houses along the road (line).

Describe an efficient algorithm that achieves this goal using as few base stations as possible. Prove the correctness of your algorithm, and analyze its time complexity.

2. (20 pts) Prove the following two properties of the Huffman encoding scheme.

- If some character occurs with frequency strictly more than \( \frac{2}{5} \), then there is guaranteed to be a codeword of length 1.
- If all the characters occur with frequency strictly less than \( \frac{1}{3} \), then no codeword of length 1 will be produced.

3. (10 pts) Suppose a server has \( n \) customers waiting to be served. The service time required by each customer is known in advance: \( t_i \) minutes for customer \( i \). So, for instance, if the customers are serviced in the order of increasing index \( i \), then the \( i \)th customer has to wait \( \sum_{j=1}^{i-1} t_j \) minutes. Our goal is to produce an order that minimizes the total waiting time:

\[
T = \sum_{i=1}^{n} \text{(time spent waiting by customer } i) \]

Give an efficient algorithm for computing the optimal order in which to service the customers. You must prove that the algorithm always finds optimal order, and also analyze its worst-case running time.

4. (10 pts) Suppose we want to use ternary codes, where each “bit” can have three values: 0, 1, 2. In other words, the tree representing the code is ternary; each node has up to three children, labeled 0, 1, and 2. Show an optimal ternary code for text that contains 4 letters with frequencies 0.1, 0.2, 0.2, 0.5? Based on this example, how would you modify Huffman’s algorithm so that it produces optimal ternary codes?