Problem 1. In a binary min-heap with \( n \) elements, both the Insert and Extract-Min operations take \( O(\log n) \) worst-case time. Give a potential function \( \Phi \) and prove that using \( \Phi \), the amortized cost of Insert is \( O(\log n) \) and the amortized cost of Extract-Min is \( O(1) \).

Problem 2. For any integer \( n > 1 \), give a sequence of operations performed on an empty Fibonacci heap \( H \), such that the resulting heap contains a single tree that is a linear chain of \( n \) nodes (that is, a tree with \( n \) nodes, and of height \( n - 1 \)).

Problem 3. We are interested in designing a data structure for maintaining a set \( A \) of integers, that supports the following operations:

- Insert\((A, x)\): Insert the integer \( x \) into the set \( A \).
- ApproximateMedian\((A)\): Return some \( x \in A \) such that at least 25% of the elements in \( A \) are not greater than \( x \) and at least 25% of the elements in \( A \) are not smaller than \( x \).

You may assume that the data structure starts with the set \( A \) being empty.

(a) Design a data structure for the above problem using a balanced binary search tree, with worst-case query time \( O(\log n) \).

(b) Design a data structure for the above problem using an array, with amortized time per query \( O(\log n) \).