1. (4 points) Exercise 17.1-2 - Show that if a DECREMENT operation were included in the k-bit counter example, n operations could cost as much as $\Theta(nk)$ time.

2. (4 points) Exercise 17.2-3 - Suppose we wish not only to increment a counter but also to reset it to zero (i.e., make all bits in it 0). Counting the time to examine or modify a bit as $\Theta(1)$, show how to implement a counter as an array of bits so that any sequence of n INCREMENT and RESET operations takes time $O(n)$ on an initially zero counter. (Hint: Keep a pointer to the high-order 1.)

3. (4 points) Exercise 19.2-1 Show the Fibonacci heap that results from calling FIB-HEAP-EXTRACT-MIN on the Fibonacci heap shown in Figure 19.4(m).

4. (5 points) Give an example of a series of insert() and extract-min operations on a Fibonacci Heap that will yield a heap of n keys with height n-1.

5. (4 points) 21.1-3 During the execution of CONNECTED-COMPONENTS on an undirected graph $G = (V, E)$ with k connected components, how many times is FIND-SET called? How many times is UNION called? Express your answers in terms of $|V|$, $|E|$, and k.

6. (4 points) Show the data structure that results and the answers returned by the FIND-SET operations in the following program. Use the linked-list representation with the weighted-union heuristic.

```plaintext
1  for i = 1 to 16
2     MAKE-SET(x_i)
3  for i = 1 to 15 by 2
4     UNION(x_i, x_{i+1})
5  for i = 1 to 13 by 4
6     UNION(x_i, x_{i+2})
7     UNION(x_1, x_5)
8     UNION(x_{11}, x_{13})
9     UNION(x_{1}, x_{10})
10    FIND-SET(x_2)
11    FIND-SET(x_9)
```