Problem 1. Show the priority queue after inserting the number 3 into the binary heap below:

```
     5
    / \
   16 17
  /   / \
 32 24 94 44
```

Problem 2. Show the priority queue after performing a delete-min on the binary heap shown in the above figure.

Problem 3. Suppose that we store the binary heap shown in Problem 1 in an array of length 9. Show the contents of the array.

Problem 4. Consider the following array $A$ of length 9. Suppose that we create a binary heap using the $O(n)$-time algorithm discussed in the class. Show the contents of $A$ at the end of the algorithm.

```
83 21 66 5 24 76 92 32 43
```

Problem 5. Show the binary heap after inserting the following integers (in this order): 10, 9, 8, 7, 6, 5, 4, 3, 2, and 1 (you should use the regular insertion algorithm).

Problem 6. Let $S$ be a dynamic set of integers. At the beginning, $S$ is empty. Then, new integers are added to it one by one, but never deleted. Let $k$ be a fixed integer. Describe an algorithm to maintain the $k$ largest integers in $S$. Your algorithm must use $O(k)$ space at all times, no matter how large $|S|$ is (note that $|S|$ increases continuously, but your space cannot). Furthermore, it must process every integer insertion in $O(\log k)$ time.

For example, suppose that $k = 3$, and that the sequence of integers inserted is 83, 21, 66, 5, 24, 76, 92, 32, 43... Your algorithm must be keeping $\{83, 66, 24\}$ after the insertion of 24, $\{83, 66, 76\}$ after the insertion of 76, and $\{83, 76, 92\}$ after the insertion of 43.