

1 Mechanical Sorting

Show the steps taken by each sort on the following unordered list:

0, 4, 2, 7, 6, 1, 3, 5

(a) Insertion sort

(b) Selection sort

(c) Merge sort

(d) Use heapsort to sort the following array (hint: draw out the heap). Draw out the array at each step:

0, 6, 2, 7, 4

2 Abstract Data Types

Recall the following ADTs when answering this question:

```
1 List
2   add(element); // adds element to the end of the list
3   add(index, element); // adds element at the given index
4   get(index); // returns element at the given index
5   size(); // returns the number of elements in the list

1 Set
2   add(element); // adds element to the collection
3   contains(object); // checks if set contains object
4   size(); // returns the number of elements in the set
5   remove(object); // removes specified object from set

1 Map
2   put(key, value); // adds key-value pair to the map
3   get(key); // returns value for the corresponding key
4   containsKey(key); // checks if map contains the specified key
5   keySet(); // returns set of all keys in map

1 Stack
2   peek(); // returns front element of stack
3   pop(); // removes and returns front element of stack
4   push(element); // adds element to front of stack

1 Queue
2   peek(); // returns front element of queue without removing it
3   poll(); // removes and returns front element of queue
4   offer(element); // adds element to front of queue

1 PriorityQueue
2   add(element); // adds element to the PQ
3   peek(); // returns front element of PQ without removing it
4   poll(); // removes and returns the highest priority element in the PQ
```

(a) For each problem, which of the ADTs given in the previous section might you use to solve each problem? Which ones will make for a better or more efficient implementation?

1. Given a news article, find the frequency of each word used in the article.

2. Given an unsorted array of integers, return the array sorted from least to greatest.

3. Implement the forward and back buttons for a web browser.

(b) Define a `Queue` class that implements the `offer` and `poll` methods of a queue ADT using only a `Stack` class which implements the stack ADT.

Hint: Consider using two stacks.