CS140 Problem Set 2, Part 1 September 27, 2017

Due date: Email your solutions to cs140_17fall@163.com or submit a written copy to a TA before 7:30 PM, October 19.

Problem 1 Illustrate the operation of radix sort on the following list of English words: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX.

Problem 2 Illustrate the operation of insertion-sort on the following sequence:
4 9 2 3 5 7 8 1 6.

Problem 3 A sequence is stored in a stack $S1$. Design an algorithm to sort the sequence using only $S1$, another stack $S2$, and a constant number of additional registers (each of which can store one input value) as storage. For each stack, you may perform the following operations.

top()	Returns the top value in the stack without modifying it
pop()	Removes the top value from the stack and returns it
push(a)	Push a value a to the top of the stack
empty()	Tests whether the stack is empty without modifying it

Problem 4 Given a random unsorted array of n numbers and $1 \leq k \leq n$, modify the Quicksort algorithm to find the k'th smallest number in expected $O(n)$ time.

Problem 5 Suppose that you are given a sorted array A of n distinct integers. Give an $O(\log(n))$ time algorithm to determine whether there exists an index i such that $A[i] = i$, For example, if $A = [-7, -1, 1, 4, 7]$, we have $A[4] = 4$. For $A = [2, 3, 4, 5, 6, 7]$, there is no such i. Give pseudocode for your algorithm and analyze its time complexity.

Problem 6 Professors Howard and Fine have proposed the following elegant sorting algorithm.

```
Function CURLY-SORT($A, i, j$):
    if $A[i] > A[j]$ then
        exchange $A[i], A[j]$;
    end
    if $(j - i + 1) > 2$ then
        $k \leftarrow floor((j - i + 1)/3)$;
        CURLY-SORT$(A, i, j - k)$;
        CURLY-SORT$(A, i + k, j)$;
        CURLY-SORT$(A, i, j - k)$;
    end
    return $A$;
```

1. Argue that, if $n = length[A]$, then CURLY-SORT(A, 1, n) correctly sorts the input array A.

2. Give a recurrence for the worst case running time of CURLY-SORT and a tight asymptotic bound on the worst case running time.

3. Compare the worst case running time of CURLY-SORT with that of insertion sort, mergesort and Quicksort. Is it better, worse or about the same as these algorithms?

The End.