CSE 100: Algorithm Design and Analysis — Release: 04-14-2020
Homework 2 — Due: 11:59pm, 4-28-2020

You must solve each problem individually – no collaboration is allowed. Submit your homework online through CatCourses – we only accept online submission. You can handwrite or type your solutions. As long as they are readable, they are acceptable.

1. (10 points) Run BUILD-MAX-HEAP on the array $A[1 \cdots 8] = \langle 3, 6, 4, 1, 2, 7, 8, 5 \rangle$. What is $A[1 \cdots 8]$ now? (Warning: if your answer is correct you will get full points. Otherwise, we can't give you partial points unless you show some intermediate snapshots of $A[1 \cdots 8]$.)

2. (10 points) We're given the following numbers as input:

   ```
   132
   232
   211
   221
   212
   111
   122
   121
   112
   ```

   Show the order of the numbers after sorting them in non-decreasing order of their last digit using the (stable) Counting-Sort. Continue to sort them by their second digit using the same algorithm and show the order.

3. (20 points) Consider a hash table with $m = 10$ slots and using the hash function $h(k) = k \bmod 10$. Say we insert (element of) keys $k = 8, 5, 15, 9, 18$ in this order. Show the final table in the following three cases.

   (a) When chaining is used to resolve collisions. Please insert the element at the beginning of the linked list.

   (b) When open addressing and linear probing are used to resolve collisions; that is, $h(k, i) = k + i \bmod 10$.

   (c) When open addressing is used with $h(k, i) = k + 2i \bmod 10$.

4. (20 points) Basic. Decision tree for Selection-Sort.

   ```
   Selection-Sort(A)
     n = A.length
       For j = 1 to n -1 Do
         smallest = j
         For i = j + 1 to n Do
            if A[i] < A[smallest] Then
               smallest = i
         Exchange A[j] with A[smallest]
   ```

(a) (12 points) Draw the decision tree corresponding to Selection-Sort when running on an array of $n = 3$ elements $A = \langle a_1, a_2, a_3 \rangle$ as in fig. 8.1 (keep "$\leq$" on the left child and "$>$" on the right child).

(b) (4 points) Mark the execution path followed for the array $A = \langle 6, 4, 2 \rangle$, as in fig. 8.1.

(c) (4 points) For the tree you drew, how many leaves does it have? Compare this with $n!$ and comment on the result.

5. (20 points) Recall the following deterministic select algorithm:

   (a) Divide the n elements into groups of size 5. So $n/5$ groups.

   (b) Find the median of each group.

   (c) Find the median x of the $n/5$ medians by a recursive call to Select.

   (d) Call Partition with x as the pivot.

   (e) Make a recursive call to Select either on the smaller elements or larger elements (depending on the situation); if the pivot is the answer we are done.

   Then, the recurrence for the running time was $T(n) = T(\frac{1}{5}n) + T(\frac{7}{10}n) + O(n)$.

   Now suppose we change the algorithm as follows.

   (a) Divide the n elements into groups of size 9. So $n/9$ groups.

   (b) Find the median of each group.

   (c) Find the median x of the $n/9$ medians by a recursive call to Select.

   (d) Call Partition with x as the pivot.

   (e) Make a recursive call to Select either on the smaller elements or larger elements (depending on the situation); if the pivot is the answer we are done.

   Now what is the **recurrence** for the running time? Also explain how much time it takes to perform **each** of the five steps.

6. (20 points) You're given n integers $a_1, a_2, ..., a_n$. You're asked to test if the n integers consist of more than k distinct integers; so k is a part of the input where $1 \leq k \leq n$. In other words, you want to test if $|\{a_1, a_2, ..., a_n\}| > k$ or not. Given a linear time algorithm (in expectation).