Exercise Sheet 7

Handout: Oct 21st — Deadline: Oct 28th - 4pm

Question 7.1 (0.25 marks)

Illustrate the operation of CountingSort$(A, B, 3)$ on the array

0	2	0	1	3	1	1

with input elements from the set $\{0, 1, 2, 3\}$.

1. For each of the first three for loops, write down the contents of array C after the loop has ended.
2. For the last for loop, write down the contents of the arrays B and C at the end of each iteration of the loop.

Question 7.2 (0.25 marks) Prove that after the **for** loop in lines 6-7 (in the pseudo-code provided at lecture) of CountingSort the array C contains in each position C[i] the number of elements less than or equal to i (You can assume the previous two **for** loops are correct).

Question 7.3 (0.25 marks) Suppose that we were to rewrite the last **for** loop header of CountingSort as

"**for** $j = 1$ to A.length".

Then the algorithm:

1. Will not be stable and will not sort the numbers
2. Will be stable but will not sort the numbers
3. Will not be stable but will sort the numbers
4. Will be stable and will sort the numbers

Justify your answer.

Question 7.4 (marks:0.5)

Describe an algorithm CountingRange(A, k, a, b) that given n integers in the range 0 to k, preprocesses its input and and then answers any query about how many integers are present in the range $[a : b]$ in $O(1)$ time. The algorithm should use $O(n + k)$ preprocessing time.

Question 7.5 (marks 0.25)

Illustrate the operation of RadixSort on the following list of English words:

COW, DOG, TUG, ROW, MOB, BOX, TAB, BAR, CAR, TAR, PIG, BIG, WOW

Question 7.6 (0.25 marks)

State which of the following algorithms are stable and which are not:

1. InsertionSort
2. MergeSort
3. HeapSort
4. QuickSort

For those that are stable argue why. For those that are not stable provide an example of an input that shows instability.

Question 7.7 (0.25 marks) Implement the 'Challenge I' and 'Nine Is Greater than Ten' on the OJ system.