CS 581 Homework 3

Due on 02/01/2018

January 26, 2018

Clarifications

- Without explicit statement, the complexity of sorting algorithm refers to its natural complexity, without any tweaks to reduce complexity in the worst cases etc.
- Without explicit statement, master theorem only refers to the three cases that were covered in class.
- n refers to the size of the problem when used in an asymptotic notation.

Problems

Problem 1.
Answer true or false, and **justify your answer**.

1) If $A = O(B)$ then $A = o(B)$.

2) If $A = O(B)$ and $A = \Omega(B)$ then $A \sim B$.

3) The worst case time complexity of merge sort is $O(n^2)$.

3) The worst case time complexity of quick sort is $O(n^2)$.

4) Two sorted lists of size m and n respectively can be merged with only $O(min(m, n))$ extra space.

5) Since merge sort is $O(n \log n)$ and insertion sort is $O(n^2)$ on average, we should always use merge sort preferably than insertion sort.

Problem 2.

Briefly answer:

1) Why would someone want to shuffle the list before applying quicksort on it?

2) Name two pivot selection strategies for quicksort other than just using the first or the last item.

3) Name one scenario where the stability of sorting matters.

4) What is the recurrence of Strassen's matrix multiplication algorithm?

Problem 3.
Given a partition scheme of quicksort below, what is the time complexity when all elements in the array are the same and when the array is already sorted? Briefly explain why for both cases.

```
algorithm quicksort(A, lo, hi) is
    if lo < hi then
        p := partition(A, lo, hi)
        quicksort(A, lo, p)
        quicksort(A, p + 1, hi)

algorithm partition(A, lo, hi) is
    pivot := A[lo]
    i := lo - 1
    j := hi + 1
    loop forever
        do
            i := i + 1
        while A[i] < pivot

        do
            j := j - 1
        while A[j] > pivot

        if i >= j then
            return j

        swap A[i] with A[j]
```

Problem 4.
Sort array {7, 3, 4, 8, 2, 1, 7, 3, 1, 6, 4, 5, 4} in non-decreasing order using counting sort. Please show your work.

Problem 5.
Sort array {789, 123, 456, 567, 234, 11, 1, 2} in non-decreasing order using radix sort. Please show your work.

Problem 6.
Use the master theorem to put asymptotic bounds on each of the following recurrences. Make the bounds as tight as possible, and justify your answer. (Don't forget the extra condition for case 3). (Hint for the fourth one: master theorem is still applicable after a little transformation. Use logarithm to convert to form $aT(n/b)$.)

1) $T(n) = 7T(n/2) + n^2 \log^2 n$
2) $T(n) = 7T(n/3) + n^2 \log^3 n$
3) $T(n) = T(n/2) + 1$
4) $T(n) = T(\sqrt{n}) + 1$

Problem 7.
Consider the regularity condition $af(n/b) \leq cf(n)$ for some constant $c < 1$, which is part of case 3 of the master theorem. Give an example of constants $a \geq 1$ and $b > 1$ and a function $f(n)$ that satisfies all the conditions in case 3 of the master theorem except the regularity condition. (Hint: It suffices to only make the regularity condition not satisfied for some values of n. Consider some $f(n)$ that is not continuous for some ns.)