**CS3364 – Design and Analysis of Algorithms Review sheet for test 1**

1. Given the flowing piece of code, in the worst case it is big-Q of what (where the input is of size n)?

Answer: N²

2. Know and understand the ordering on different rates of growth on different standard functions (i.e. lg n grows slower than n, n5 grows slower that 2n, etc.)

Answer: logN < Constant < NlogN < N < N² < N!

3. Under what situation might bubble sort run faster than quicksort on a given data set?

Answer: For small inputs.

4. Given the following data stored in an array in the order given, what would the contents of the array look like after the first application of the partition routine from quicksort used in the book?

17, 42, 12, 8, 34, 16, 9, 4, 22, 43, 13, 19

Answer: 17,12,8,16,9,4,13,19,22,43,34,42

5. Given the initial data in the order below, show the data after each of the 3 iteration of the radix sort routine:

234, 526, 278, 129, 312, 221, 193, 426

Answer:

Initially: 234 526 278 129 312 221 193 426

Sort on right-most digit: 221 312 193 234 526 426 278 129

Sort on middle digit: 312 221 526 426 129 234 278 193

Sort on leftmost digit: 129 193 221 234 278 312 426 526

6. The Master Method has the following 3 cases:

1. if f(n) = O(n(logba)-e) for some constant e > 0, then T(n) = Q(nlogba)  
2. if f(n) = Q(nlogba), then T(n) = Q(nlogba lg n)  
3. if f(n) = W(n(logba)+e) for some constant e > 0 and if af(n/b) ≤ cf(n) for some constant c > 1 and all sufficiently large n, then T(n) = Q(f(n)).

What is the asymptotic bounds of T(n) = 8T(n/2) + n?

Answer: since a=8 b=2 then log28 = 3, because f(n)=n, so this’s case 1, the answer would be T(n) = Q(n3).

7. Describe how the max-heapify routine from heapsort works (make sure to state what assumptions are made on the input).

Answer: max-heapify assumes we have a binary tree where the left and right subtrees of the root are both max-heaps but the root may not be in the right position for the whole tree to be a max heap. The algorithm works as follows:

1)  start with the current node being the root of the whole tree

2)  if the current node is a leaf or its data is bigger than the data of both of its children

then stop,

3) otherwise swap the data in the current node with the data in the bigger of the two children and go back to step 2) with the current node being the child you swapped with.

8. Describe counting sort. When is counting applicable?

Answer: Counting sort is applicable when all the values in the array being sorted are between the value of 0 and some K (inclusive). The algorithm then works as follows:

1) Create and array, A, from 0 to K which is initially all 0s.  
2) Loop through input and for each item i add 1 to A[i].  
3) (without going into details of books implementation) Create a new output array

and loop through array A, putting A[i] i’s into the output.

9. Other than searching, sorting, and the max sub-array problem given in class, given another problem for which a divide and conquer algorithm would be applicable.

Answer: given a continuous mathematical function, f(X), and two values a and b where f(a) > 0 and f(b) < 0 (or vice versa), the problem of finding a value, c, such that f(c) = 0 can be found by a divide and conquer algorithm.

10. Write the code for a sort routine that is O(n2). You may use pseudocode so long as the control flow is understandable.

Answer: Selection sort: Given a input list of length n, first create an output list which is initially empty, next go through the input list n times doing the following 2 steps: 1) use a linear search to find the smallest item in the input list 2) remove the smallest item from the input list and append it to the output list

11. What properties should a good hash function have?

Answer: first, the function should run quickly. Second, the hush function should put the data into bins with a roughly even distribution.