# CS251 Homework 2: Sorting and Hashing

**Out: Feb 2, 2018 @ 9:00 PM**

**Due: Feb 9, 2018 @ 11:59 PM**

**Submission Instructions: Please submit a typeset PDF on blackboard.**

## Short-Answer Questions:

C 1. What are the worst-case time complexities of 'insert' and 'find' operations in a dictionary ADT, where n is the size of the dictionary?

a) O(1) & O(1)
b) O(n) & O(n)
c) O(1) & O(n)
d) O(n) & O(1)

D 2. Which of the following sorting algorithms has the lowest worst-case time complexity?

a) Selection sort
b) Insertion sort
c) Bubble sort
d) Heap sort

A 3. Which of the following sorting algorithms is not stable (i.e., the initial order of equal keys is maintained)?

a) Selection sort
b) Insertion sort
c) Bubble sort
d) All sorting algorithms are stable

4. Suppose you have the following hash table, implemented using linear probing. The hash function we are using is the identity function, h(x) = x % 7.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 14 | 21 | 9 | 3 | 4 | 12 | 18 |

Which of the following is/are a possible order of insertion into the hash table? There may be multiple correct answers. Select all that apply.

a) 9, 14, 4, 18, 12, 3, 21
b) 12, 3, 14, 18, 4, 9, 21
c) 12, 14, 3, 9, 4, 18, 21
d) 9, 12, 14, 3, 4, 21, 18
e) 12, 9, 18, 3, 14, 21, 4

5. What is the worst case running time of search with linear probing?
    a) O(1)
    b) O(n)
    c) O(log n)
    d) O($\sqrt{n}$)

6. What are the 'expected running time of insertion' and 'worst case running time of deletion', in a hash table with quadratic probing to resolve collisions?
    a) O(log n) and O($\sqrt{n}$)
    b) O(1) and O(n)
    c) O($\sqrt{n}$) and O($\sqrt{n}$)
    d) O($\sqrt{n}$) and O(n)

7. Using alphabet position, encode 'science' with the following hash map
    p (z) = $a_0 + a_1 z + a_2 z^2 + \ldots + a_{n-1} z^{n-1}$ with z=19

8. Which of the following sorting algorithms takes linear time in the best case?
    a. Selection sort
    b. Insertion sort
    c. Bubble sort
    d. Heap sort

9. A heap with n keys has a height of:
    a. $\Theta$(n)
    b. $\Theta$(log n)
    c. $\Theta$(n log n)
    d. $\Theta(\log^2 n)$

10. Time complexity of getMin() and removeMin() operations on a min-heap are respectively
    a. $\Theta$(1) & $\Theta$(log n)
    b. $\Theta$(log n) & $\Theta$(log n)
    c. $\Theta$(log n) & $\Theta$(1)
    d. $\Theta$(1) & $\Theta$(n)

**Long-Answer Questions:**

11. Given two sets A and B represented as sorted sequences, describe an efficient algorithm for computing $A \oplus B$, which is the set of elements that are in A or B, but not in both.

12. Let S be a random permutation of n distinct integers. Argue that the expected running time of insertion-sort on S is $\Omega(n^2)$.
(Hint: Note that half of the elements ranked in the top half of a sorted version of S are expected to be in the first half of S.)

13. Come up a compression map (function) that results in no collisions for the following data for a 7-entry hash table: 55, 45, 12, 1, 4, 23, 7. Draw the resulting hash table.

14. Draw the 7-entry hash table that results from hash function, h(i) = (2i+5)%7, to hash the keys 12, 34, 13, 78, 21, 91, 11, 39, 10, 6, and 15, assuming collisions are handled by chaining.

15. Does a preorder traversal of a heap list its keys in non-decreasing order? Argue that it is right or give a counterexample to disprove.