CS 61B Sorting and Hashing Fall 2014

1 Sorting I

Show the steps taken by each sort on the following unordered list:

`106, 351, 214, 873, 615, 172, 333, 564`

(a) Quicksort (assume the pivot is always the first item in the sublist being sorted and the array is sorted in place). At every step circle everything that will be a pivot on the next step and box all previous pivots.

(b) Merge sort. Show how the list is broken up at every step.

(c) LSD radix sort.

(d) Give an example of a situation where using insertion sort is more efficient than using merge sort.

2 Sorting II

Match the sorting algorithms to the sequences, each of which represents several intermediate steps in the sorting of an array of integers.

Algorithms: Quicksort, merge sort, heap sort, MSD radix sort, insertion sort.

(a)
```
12, 7, 8, 4,  10, 2,  5,  34, 14
7,  8, 4, 10, 2,  5,  12, 34, 14
4,  2, 5, 7,  8,  10, 12, 14, 34
```

(b)
```
23, 45, 12, 4, 65, 34, 20, 43
12, 23, 45, 4, 65, 34, 20, 43
```

(c)
```
12, 32, 14, 11, 17, 38, 23, 34
12, 14, 11, 17, 23, 32, 38, 34
```

(d)
```
45, 23, 5,  65, 34, 3,  76, 25
23, 45, 5,  65, 3,  34, 25, 76
5,  23, 45, 65, 3,  25, 34, 76
```

(e)
```
23, 44, 12, 11, 54, 33, 1, 41
54, 44, 33, 41, 23, 12, 1, 11
44, 41, 33, 11, 23, 12, 1, 54
```

3 Hash Codes

(a) Suppose that we represent Tic-Tac-Toe boards as 3 by 3 arrays of integers (each of which is in the range 0 to 2). Describe a good hash function for Tic-Tac-Toe boards that are represented in this manner. Try to come up with one such that boards that are not equal will never have the same hash code.

(b) Is it possible to add arbitrarily many Strings to a Java HashSet with no collisions? If not, what is the minimum number of distinct Strings you need to add to a HashSet to guarantee a collision?

4 Bonus Question

Describe a way to implement a linked list of Strings so that removing a String from the list takes constant time. You may assume that the list will never contain duplicates.