107-1 PDSA Midterm

Nov. 5, 2018

學號：_____

姓名：_____

1. (5%) Provide the content of the array that results after the first 6 exchanges (not iterations!) when applying insertion sort to the following array:

   15 31 43 60 88 23 62 94 72 69

2. (5%) Provide the content of the array that results after applying the standard 2-way partitioning algorithm from lecture to the following array:

   74 89 23 46 63 94 40 13 16 38 20 81

Recall, in the standard 2-way partitioning algorithm, the leftmost entry is the partitioning item.

3. (10%) Which of the following statements about sorting algorithms are true? Check all that apply.
   (a) Mergesort is a stable sorting algorithm.
   (b) Quicksort is an inplace sorting algorithm.
   (c) Heapsort is a stable sorting algorithm.
   (d) The number of compares to 3-way quicksort an array of N distinct keys is linear in the best case.
   (e) No key can be involved in more than ~ 1/2 N compares when applying Mergesort to an array containing N distinct keys.

4. (5%) Provide the sequence of the 13 keys in the array that results after inserting the sequence of 3 keys: 76 16 92

into the following maximum-oriented binary heap of size 10:

86 82 77 75 70 35 68 31 45 30

5. (5%) Provide the sequence of the 7 keys in the array that results after performing 3 successive delete-the-max operations on the following maximum-oriented binary heap of size 10:

   97 82 89 34 66 78 85 15 28 51

6. (10%) Which of the following id[] array(s) could be the result of running the weighted quick union algorithm on a set of 10 items? Check all that apply.

   (a) 8 0 2 3 0 8 5 6 3 5
   (b) 9 5 4 6 4 4 4 4 4 5
   (c) 4 4 6 4 5 2 4 4 4 6
   (d) 0 2 2 3 4 5 0 7 9 9
   (e) 6 9 9 6 0 2 9 0 9 9

Recall: our weighted quick union algorithm uses union by size (number of nodes), not union by height.

7. (10%) Analyze the complexity of the following function by first writing down its recurrence relation and then solving it.

```
private static void algorithm(int n) {
   if(n<=1) return;
   for(int i=1; i<n; i++) n = n;
   for(int i=0; i<4; i++) {
      algorithm(n/3);
   }
}
```

8. (25%) Please implement the three functions ('isEmpty', 'enqueue' and 'removeFirst') of CircularQueue.java with the following API:

```
public class CircularQueue <Item> implements Iterable {
   private Node last;
   private class Node{
      Item item;
      Node next;
   }
   ...
   public boolean isEmpty() { // Is the circular queue empty?
   // Provide your code here
   }
   public void enqueue(Item item) { // Add item to the end of the
list
   // Provide your code here
   }
   public Item removeFirst() { // Remove item from the beginning
of the list
   // Provide your code here
   }
   ...
}
```

In a circular queue, no links are null and the value of last.next is first whenever the list is not empty. Keep only one Node instance variable (last).

9. (10%) Calculate the number of compares C(N) when applying Mergesort to an array of size N.

10. (15%) Write a recursive function that converts infix expressions to postfix.

```
public String infix2postfix (String e) {
    private String s;
    // Provide your code here
    return s;
}
```

Example of input: (1 + ((2 + 3) * (4 * 5)))

Example of output: 1 2 3 + 4 5 * * +

Note: All the numbers, operators, and brackets are separated by 'single space', and each bracket contains one operator and two operands only.