Final Exam in Algorithms and Data Structures

Department of Information Technology
Uppsala University
2010–10–15
Lecturers: Parosh Aziz Abdulla and Jonathan Cederberg
Location: Gimogatan
Time: $8 - 13$
No books or calculator allowed

Directions:

1. Answer only one problem on each sheet of paper
2. Do not write on the back of the paper
3. Write your name on each sheet of paper
4. **Important** Unless explicitly stated otherwise, justify you answer carefully!! Answers without justification do not give any credits.

Good Luck!

Problem 1 (10 p) List the following functions by increasing asymptotic growth rate. If two functions have the same asymptotic growth rate, state that fact. No justification is needed.

$$2\lg n \qquad n\lg n \qquad \lg(n^2) \qquad n^2 \qquad n^3 \qquad 3\lg n \qquad 2^5\lg n \qquad \left(\frac{1000}{999}\right)^n$$

Problem 2 (15 p) Consider the *QuickSort* algorithm. Suppose that the length of the input array is odd. Furthermore, suppose that the elements are initially ordered so that the one in the middle is the largest one, then ones immediately to the left and the right of the middle element are equal and the next largest ones, and so on. The leftmost and the rightmost elements are consequently the smallest ones. Examples of such initial configurations are give below:

$$1 \quad 2 \quad 4 \quad 6 \quad 8 \quad 9 \quad 8 \quad 6 \quad 4 \quad 2 \quad 1$$

$$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 8 \quad 12 \quad 20 \quad 12 \quad 8 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1$$

What will be the performance of the algorithm in such a case?
Remark Your answer should not be longer than **3 lines**.

Problem 3 (15 p) Simulate *MergeSort* on the following array:

$$[8\ ,\ 4\ ,\ 3\ ,\ 5\ ,\ 6\ ,\ 2\ ,\ 4\ ,\ 10\ ,\ 4\ ,\ 8\ ,\ 12\ ,\ 7]$$

Your presentation should show all intermediate arrays created, and the relationship between them. You do not have to show the array state in the middle of loops.

Problem 4 (10 p) Given below is the original code for counting sort. Suppose you changed line 9, to traverse A forward instead of backward. What effect would this have on the algorithm? Your answer should be no longer than three lines.

COUNTING-SORT(A, B, k)

```
 1  for i ← 0 to k
 2        do C[i] ← 0
 3  for j ← 1 to length[A]
 4        do C[A[j]] ← C[A[j]] + 1
 5  ▷ C[i] now contains # elements = i.
 6  for i ← 1 to k
 7        do C[i] ← C[i] + C[i − 1]
 8  ▷ C[i] now contains # elements ≤ i.
 9  for j ← length[A] downto 1
10        do B[C[A[j]]] ← A[j]
11           C[A[j]] ← C[A[j]] − 1
```

Problem 5 (15 p) Consider the code for computing the successor of a node in a binary search tree.

TREE-SUCCESSOR(x)

```
1  if right[x] ≠ NIL
2     then return  TREE-MINIMUM(right[x])
3  y ← p[x]
4  while y ≠ NIL and x = right[y]
5        do x ← y
6           y ← p[y]
7  return (y)
```

- Is the **while** loop of line 4 guaranteed to terminate? (answer *yes* or *no*).
- For what types of trees will the loop terminate because the condition $y \neq NIL$ has failed; and for what types of trees will the loop terminate because the condition $x = right[y]$ has failed? (answer max 2 lines)
- Can it happen that both conditions fail at the same time? if *yes*, describe when; if *no*, explain why? (answer max 2 lines)

Problem 6 (10 p) Show the result of inserting the following sequence of numbers from left to right in an initially empty max-heap:

7 1 3 15 6 2 4 20 1 8

Problem 7 (10 p) Consider the following four hash functions, and a hash table of size $m > 100$.

- $h_1(k) = k \mod m$
- $h_2(k) = 2k \mod m$
- $h_3(k) = \lfloor \frac{k}{100} \rfloor \mod m$
- $h_4(k) = k \cdot m \mod m$

Which hash function(s) is (are) the best to use if

a) all keys we want to hash are in the set $\{0, 1, 2, \ldots, 999\}$.

b) keys are arbitrarily large, and m is odd.

c) keys are even and arbitrarily large, and m is even.

Justify each answer in one sentence.
Remark There might be more than one best function. In such a case, list all of them.

Problem 8 (15 p) Consider the depth-first search algorithm for graphs (DFS). Suppose that at some point during the run of the algorithm, a node v is red (discovered but not finished), and a node u is white (not yet discovered). Also assume that there is a path from v to u. Can it happen that v becomes blue before u does? If *yes*, draw a graph (no more then six nodes) in which this can happen. If *no*, explain the reason in no more than three lines.