Final Exam in Algorithms and Data Structures 1 (1DL210)

Department of Information Technology
Uppsala University
2011-10-14
Lecturers: Parosh Aziz Abdulla, Jonathan Cederberg and Jari Stenman
Location: Polacksbacken
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 (15 p)

a) 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.

$$\lg(2n) \qquad 2n \qquad 2^n \qquad 2^{n+1} \qquad (2n)^2 \qquad n^2 \qquad 2\lg n \qquad 2^{(2n)}$$

b) State whether the following statements are true or false. No explanation is needed.

(i) If $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$, then we have $f(n) = \Theta(g(n))$

(ii) Searching for an element in a linked list is logarithmic in the size of the list.

Problem 2 (5p) Describe the concept of "divide-and-conquer" in two sentences.

Problem 3 (10 p) We know that RADIX-SORT has complexity $\mathcal{O}(d(n+k))$. State in one sentence

a) what d stands for.

b) what n stands for.

c) what k stands for.

Also explain in at most 3 sentences

d) what assumptions we make to get RADIX-SORT as $\mathcal{O}(d(n+k))$.

Problem 4 (10 p) Assume that we are given an array the objects where each object is a pair of keys (k_1, k_2). The objects are already sorted according to their first keys. Furthermore, the value of the second key k_2 for each object is close to the value of the first key k_1. Assume that we now want to sort the objects according to their second keys instead.
Example: The input array consists of the elements

$(223, 250)$ $(433, 430)$ $(501, 508)$ $(622, 625)$ $(630, 624)$ $(800, 810)$ $(944, 942)$

The result of sorting the array is

$(223, 250)$ $(433, 430)$ $(501, 508)$ $(630, 624)$ $(622, 625)$ $(800, 810)$ $(944, 942)$

Would you use the QUICKSORT algorithm for this purpose? Your answer should start with one of the words *yes* or *no*. Motivate you answer in no more than 5 sentences.

Problem 5 (15p) As we saw in Problem 2, the runtime of an algorithm can depend in things other than just the size of the input. Consider the code for INSERTION-SORT, given below. Now assume that your elements in the array are strings. For simplicity, assume all strings to be of the same length s. Also assume that the comparison $A[i] > key$ on line 5 is not a constant time operation, but instead linear in the size of the strings compared, i.e. it is $\mathcal{O}(s)$.
Derive an asymptotically tight upper bound on the worst case of INSERTION-SORT under these new circumstances. That is, find a function $f(n, s)$ such that INSERTION-SORT is $\mathcal{O}(f(n, s))$.
Justify your asymptotic bound in 3 sentences.

INSERTION-SORT(A)

```
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into A[1 . . j − 1].
4          i ← j − 1
5          while i > 0 and A[i] > key
6               do A[i + 1] ← A[i]
7                  i ← i − 1
8          A[i + 1] ← key
```

Problem 6 (20p) Suppose you have a set of n numbers where you over some time period do $\lg n$ searches and $\lg n$ lookups of the maximum. To maintain your set of numbers, you you can choose between using either a heap or a binary search tree.

a) Assume you are interested in minimizing the average total runtime over the time period. For each structure, what is the asymptotic upper bound on the total runtime, and which of the two structures should you choose?

b) Suppose you realize that you are actually interested in the worst case. For each structure, what is the asymptotic upper bound on the worst case runtime, and which of the two structures should you choose now?

Each justification should be no longer than three sentences.

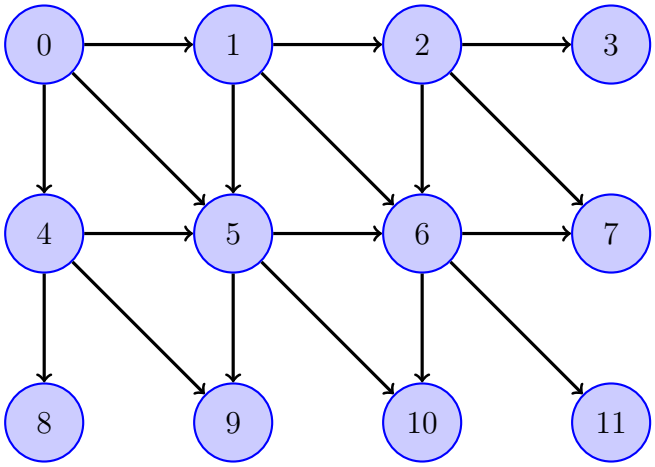

Figure 1: The graph G

Problem 7 (15p)

a) Suppose we do a Breadth-First-Search starting at node 0 in the graph G. Can it happen that we discover (color red) node 7 before we discover node 9? Justify the answer in 2 sentences.

b) If we do a Depth-First-Search starting at node 0 in G, can it happen that we discover (color red) node 10 before we discover node 5? Justify the answer in 2 sentences.

c) Give two topological sortings of G. No justification needed.

Problem 8 (10p)

Assume you insert the sequence $4, 3$ into an initially empty binary search tree, and then the numbers $1, 2, 5, 6, 7$ in some order. What is the maximum height (number of levels) the final tree can have? Justify your answer in at most three sentences.