

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

Department of Information Technology

Uppsala University

February 30th, 2012

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 your answer carefully!!
Answers without justification do not give any credits.

Good Luck!

Problem 1 (20 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.

$$\lg n \quad 1.1^n \quad n \lg n \quad n(\lg n)^2 \quad 3 \lg n \quad 2^5 \quad n^{34}$$

- 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 $(f(n))^2 = \Theta((g(n))^2)$
- (ii) If $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$, then we have $f(n) = g(n)$
- (iii) $2^n + n^2 = \mathcal{O}(3^n)$.
- (iv) $2^n + n^2 = \mathcal{O}(2^n)$.

Problem 2 (15p) Consider INSERTION-SORT and MERGE-SORT. For each algorithm, what will be the worst case asymptotic upper bound on the running time if you know additionally that

- a) the input is already sorted?
- b) the input is reversely sorted?
- c) the input is a list containing n copies of the same number?

For each case and each sorting algorithm, state your answer and justify it in one sentence.

Problem 3 (20p) Suppose you have a set of numbers where you over some time period do a constant number of insertions and a linear number of lookups of the maximum. To maintain your set of numbers, 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 average case 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 two sentences.

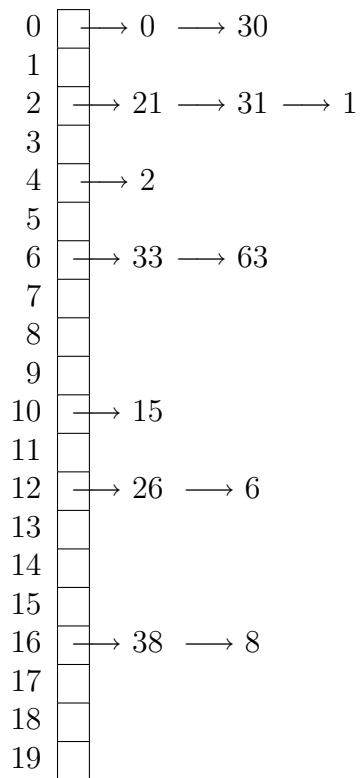
Problem 4 (10p) We know that hash tables are very common in practice when you have a lot of insertions, deletions and search operations. We have seen that under the assumption of uniform hashing, collision resolution by chaining, and constant time computable hash function, deletion and search are both $\mathcal{O}(1 + \frac{n}{m})$ in the average case.

We have also seen that this can in fact be reduced to $\mathcal{O}(1)$ under one additional assumption. Describe this additional assumption in one sentence.

Problem 5 (15p) Suppose you have a hash table and have inserted some elements. When you inspect it you see that the result looks as the picture below. Being a good student, you realize this is a problem.

- What is the problem here?
- Give an example of a hash function that could give rise to this behavior.
- What would be a better hash function?

Answer each question in one sentence.



Problem 6 (10p)

Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 45. Which (possibly multiple) of the following sequences could be the sequence of nodes examined?

- 5, 2, 1, 10, 39, 34, 77, 63.
- 1, 2, 3, 4, 5, 6, 7, 8.
- 9, 8, 63, 0, 4, 3, 2, 1.
- 8, 7, 6, 5, 4, 3, 2, 1.
- 50, 25, 26, 27, 40, 44, 42.
- 50, 25, 26, 27, 40, 44.

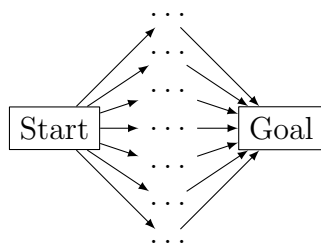
Problem 7 (10p) Suppose that we first insert an element x into a binary search tree that does not already contain x . Suppose that we then immediately delete x from the tree. Will the new tree be identical to the original one? If *yes* give the reason in no more than 3 sentences. If *no* give a counter-example. Draw pictures if you necessary.

Problem 8 (10p) Suppose we have a heap H and two values v_1 and v_2 , such that all values are distinct. Let H_{12} be the heap you get if you insert v_1 and then v_2 into H , and H_{21} be the heap you get if you insert v_2 and then v_1 into H . Give an example of H , v_1 and v_2 such that $H_{12} \neq H_{21}$. No justification needed, just draw the heaps H , H_{12} and H_{21} .

Problem 9 (10p)

Suppose you have a graph like the one below. The dots signify some part of the graph that you don't know exactly, *not* a straight link. You know however, that there are *many* paths from the start to the goal. You also know that they tend to be rather long. Suppose that you want to implement a program that searches for a path and returns the first one it can find. You have no need for finding the optimal path in any sense, just any path will do.

Would you want to use BFS or DFS as the basis for your program? Justify your answer in at most two sentences.



Problem 10 (10p) Give one possible DFS traversal starting from node 0 of the graph below, printing the nodes both as they are discovered and finished. Note that the graph is **directed**; for instance, there is an edge from 8 to 4 but not from 4 to 8.

