Introduction to Algorithms and Datastructures

IT University of Copenhagen

9. januar 2002

This exam consists of 3 exercises containing in total 13 subexercises. Each of these 13 subexercises are given the same weight in the evaluation. You have 4 hours to complete the exam. Remember to number the pages and write your name and CPR number on every page. The exam consists of 4 numbered pages. CLR refers to "Introduction to Algorithms" by Cormen, Leiserson and Rivest, 18. print, 1997. CLRS refers to "Introduction to Algorithms" Second Editon by Cormen, Leiserson, Rivest and Stein, 2001.
For exercises, in which effective algorithms must be specified, the asymptotic time complexity of the specified solution will be taken into account when grading. Exercises asking for time complexity must be answered using O-notation with least possible asymptotic growth.

Exercise 1

This exercise is about determining O-functions and sorting from the chapters 2 and 8 in CLRS, or correspondingly the chapters 1 and 9 in CLR.

Look at the code A:

```
1 for i ← 1 to n
2     do A[i] ← i
3        B[i] ← 1
4 MERGE-SORT(A, 1, n)
5 COUNTING-SORT(A, B, n)
6 for i ← 1 to n
7     do COUNTING-SORT(A, B, n)
```

a) Give the time complexity for executing the lines 1-5 in the code A.

b) Give the time complexity for executing the lines 6-7 in the code A.

Look at the procedure ADD1(A, m)

```
1 if m ≥ 2
2     then for i ← 1 to m
3              do A[i] ← A[i] + 1
4          ADD1(A, ⌊m/2⌋)
```

Look at the code B:

```
1 for i ← 1 to n
2     do A[i] ← 0
3 ADD1 (A, n)
```

c) Give the time complexity for executing ADD1 i linie 3 for koden B.

d) Using O-notation, denote the largest value in the array A after executing ADD1 in line 3 of the code B.

Exercise 2

This exercise is about graphs and minimum spanning trees with terminology as defined in chapter 22 and 23 of CLRS, or chapter 23 and 24 in CLR correspondingly. Look at the following unoriented graph

$$H = (\{1, 2, 3, 4, 5\}, \{(1, 2), (2, 5), (4, 5), (2, 3), (1, 5), (3, 5), (3, 4)\}).$$

a) Draw the graph H.

b) Give the adjacency representation for H as in figure 22.1 page 528 in CLRS or correspondingly figure 23.1 page 466 in CLR.

c) Denote edges crossing the cut $(\{1, 2\}, \{3, 4, 5\})$ in H.

Let $G = (V, E)$ be a connected, unoriented, weighted graph with n vertices and m edges. Let the set of vertices V be defined as the integers $\{1, 2, \ldots, n\}$ as usually.

d) Describe an effective algorithm which, given G in adjacency-list representation, fills a table A with n entries, where each entry i contains a j such that the edge (i, j) is part of a minimum spanning tree for G. That is; for all vertices $i \in V$ it is the case that$(i, A[i])$ is an edge in a minimum spanning tree for G. Provide the time complexity of your solution.

e) Describe an effective algorithm which, givem G in adjacency-list representation and an edge $e \in E$, decides whether e is part of some minimum spanning tree of G. Provide the time complexity of your solution.

Exercise 3

This exercise is about the datastructure for disjoint sets as defined in chapter 21 in CLRS or chapter 22 in CLR, as well as binary search trees.

Look at the following code C:

```
1 for i ← 1 to 5
2     do MAKE-SET(i)
3 UNION(1, 3)
4 UNION(2, 3)
5 UNION(5, 3)
```

a) Is it the case that FIND-SET(2)=FIND-SET(5) after executing line 1-5 in the code C?

Look at the following code D:

```
for i ← 1 to n
    do MAKE-SET(i)
for i ← 2 to n
    do UNION(1, i)
```

b) Assume that the above union operations are implemented as *disjoint-set forests* (CLRS chapter 21.3 and coorespondingly CLR chapter 22.3) and with the *union by rank* heuristic. What is the *worst-case* time complexity for one UNION operation in the sequence of operations executed in the above code D?

c) Let A be a table with n integers, orderet in ascending order. Describe an effective algorithm, which builds a balanced binary searchtree for the numbers in A. That is; the height must be $O(\log n)$. Provide the time complexity of your solution.

d) Let A be a table with n integers between 1 and $3n$. Describe an effective datastructure, which saves the integers in A in linear space in such a way that the datastructure afterwards can support the operation:
SEARCH(k): Returns `true` if the integer k is in A and `false` otherwise.
Provide the time complexity for building the datastructure and SEARCH.