Self-Study Session 01

Algorithms and Data Structures (DAT2, SW2, DV2)

Instructions. You have to **work individually** on these questions from 8:15 to 10:00. From 10:00 to 12:00 you can join your group and discuss your solutions togther.

- Read carefully the text of each exercise before solving it! Pay particular attentions to the terms in bold.
- If a question seems ambiguous, write under which assumptions you are solving it.
- You can use **CLRS** to refer to the textbook T.H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms* (3rd edition) in your answers.
- Make an effort to use a readable handwriting and to present your solutions neatly.
- The TAs will be available from 10:00 to 12:00. Use the digital trashcan to ask for help.
- The points relative to each question give an indication of their complexity relative to this exercise sheet. These points **may not** correspond to the amount of points you may get for a similar exercise at the exam.

Question 1. 20 Pts

Identifying asymptotic notation. (Note: lg means logarithm in base 2)

(1.1) [5 Pts] Mark **ALL** the correct answers. $\lg n^2 + \lg 2^n + 1^n + \sqrt{n}$ is

☐ **b)** $\Theta(\lg n)$ ☐ **b)** $\Theta(n)$ ☐ **c)** $\Theta(\sqrt{n})$ ☐ **d)** $\Theta(n^2)$ ☐ **e)** $\Theta(1^n)$

(1.2) [5 Pts] Mark **ALL** the correct answers. $\lg n^2 + \lg 2^n + 1^n + \sqrt{n}$ is

☐ **a)** $O(\lg n)$ ☐ **b)** $O(n)$ ☐ **c)** $O(\sqrt{n})$ ☐ **d)** $O(n^2)$ ☐ **e)** $O(1^n)$

(1.3) [5 Pts] Mark **ALL** the correct answers. $558 \cdot n \lg n^2 + 0.0001 \cdot n^3 + (n \lg n)^2$ is:

☐ **a)** $\Theta(n \lg n)$ ☐ **b)** $\Theta(n^3)$ ☐ **c)** $\Theta(n^2)$ ☐ **d)** $\Theta(n^2 \lg n)$ ☐ **d)** $\Omega(n^2 \lg n)$

(1.4) [5 Pts] Mark **ALL** all the functions below that satisfy $\lg(f(n)) = \Theta(\lg n)$.

a) **a)** $f(n) = n^2$ b) **b)** $f(n) = 2^n$ c) **c)** $f(n) = 2^n \cdot n^2$ d) **d)** $f(n) = \sum_{i=1}^{n} i$

Question 2. 20 Pts

Consider the following recurrences

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(n/2) + n^4 & \text{if } n > 1 \end{cases} \qquad Q(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 9 \cdot Q(n/3) + n^{1.9} & \text{if } n > 1 \end{cases}$$

(2.1) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** $T(n)$ can be solved using Case 1 of the Master Theorem
- ☐ **b)** $T(n)$ can be solved using Case 2 of the Master Theorem
- ☐ **c)** $T(n)$ can be solved using Case 3 of the Master Theorem
- ☐ **d)** $T(n)$ cannot be solved using the Master Theorem

(2.2) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** $T(n) = \Theta(n^4 \lg n)$
- ☐ **b)** $T(n) = \Theta(n^2)$
- ☐ **c)** $T(n) = \Omega(n^3)$
- ☐ **d)** $T(n) = O(n^4)$

(2.3) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** The recurrence Q is in the form $Q(n) = a\,Q(n/b) + f(n)$ for some a, b and $f(n)$.
- ☐ **b)** $T(n)$ can be solved using Case 1 of the Master Theorem
- ☐ **c)** $T(n)$ can be solved using Case 4 of the Master Theorem
- ☐ **d)** $T(n)$ cannot be solved using the Master Theorem

(2.4) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** $Q(n) = \Theta(n^{1.9} \lg n)$
- ☐ **b)** $Q(n) = \Omega(n^{1.9})$
- ☐ **c)** $Q(n) = \Theta(n^2)$
- ☐ **d)** $Q(n) = O(n^3)$

Question 3. 20 Pts

(3.1) [5 Pts] Consider the INSERTION-SORT algorithm on the following input $A = [91, 71, 29, 43, 97, 59, 17, 93, 61, 13]$. Select among the followings, the configuration of the array after **four** iterations of the main loop of the INSERTION-SORT algorithm have been performed.

☐ **a)** $[29, 43, 71, 91, 97, 59, 17, 93, 61, 13]$ ☐ **b)** $[3, 17, 29, 43, 59, 61, 71, 91, 97, 93]$

☐ **c)** $[71, 29, 43, 59, 17, 61, 13, 91, 97, 93]$ ☐ **d)** $[29, 43, 91, 71, 97, 59, 17, 93, 61, 13]$

☐ **e)** None of the above is correct.

(3.2) [5 Pts] Consider an execution of MERGE-SORT$(A, 1, A.\mathit{length})$ on the array $A = [7, 3, 5, 9, 8, 2]$. Select among the following options, the configuration of the array after **three** calls of the subroutine MERGE have been performed.

☐ **a)** $[5, 3, 7, 9, 8, 2]$ ☐ **b)** $[7, 3, 5, 9, 8, 2]$ ☐ **c)** $[3, 5, 7, 8, 9, 2]$

☐ **d)** $[2, 3, 5, 7, 8, 9]$ ☐ **e)** None of the above is correct.

(3.3) Consider the algorithm STACKSTUFF, which takes as input an integer $n > 2$ and performs some operations on a stack S which is initially empty.

```
StackStuff(n)
Initialise an empty stack S
for i = 1 to n − 2
    for j = n downto i
        Push(S,i)
for k = 1 to n^2
    Pop(S)
```

(a) [5 Pts] What is the asymptotic running time of STACKSTUFF(n)? Justify your answer.

(b) [5 Pts] What is the size of the stack S after the execution of STACKSTUFF(n)?

☐ **a)** $\Theta(1)$ ☐ **b)** $\Theta(n)$ ☐ **c)** $\Theta(n \lg n)$ ☐ **d)** $\Theta(n^2)$ ☐ **e)** $\Theta(n^3)$

Question 4. 40 Pts

The Fibonacci sequence is the series of numbers $1, 1, 2, 3, 5, 8, 13, 21, \ldots$ where the next number in the sequence is found by adding up the two numbers before it. The n-th Fibonacci number in the sequence is typically defined by the following recurrence (see CLRS pp. 99) for $n \in \mathbb{N}$

$$F(n) = \begin{cases} 1 & \text{if } n \in \{0, 1\}, \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

(a) [15 Pts] Consider the following iterative algorithm FIB-ITER(n) which computes the n-th Fibonacci number. Prove that FIB-ITER(n) is correct.

```
FIB-ITER(n)
Initialise the array F[1 . . n + 1]
F[1] = 1
F[2] = 1
for i = 3 to n + 1
    F[i] = F[i − 2] + F[i − 1]
return F[n + 1]
```

(b) [5 Pts] What is the asymptotic running time of FIB-ITER(n)? Justify your answer.

(c) [20 Pts] Consider the recursive algorithm FIB-REC(n) which computes $F(n)$ by implementing its recurrence. Prove that the running time of FIB-REC(n) is $O(2^n)$.

```
FIB-REC(n)
if n < 2
    return 1
else
    return FIB-REC(n − 1) + FIB-REC(n − 2)
```