Algorithms and Data Structures (DAT3/SW3)

Exam Assignments

Bin Yang & Chenjuan Guo

11 January 2019

Full name:	
Student number:	
E-mail at student.aau.dk:	

This exam consists of three exercises and there are **three** hours (10.00 a.m. to 13.00 p.m.) to solve them. When answering the questions in exercise 1, mark the check-boxes on this paper. Remember also to put your name and your student number on any additional sheets of paper you will use for exercises 2 and 3.

During the exam you are allowed to consult books and written notes. However, the use of any kind of electronic devices with communication functionalities, e.g., laptops, tablets, and mobile phones, is **NOT** permitted. You can bring old fashion calculators.

- *Read* carefully *the text of each exercise before solving it! Pay particular attentions to the terms in* **bold**.
- **CLRS** *refers to the textbook—T.H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein,* Introduction to Algorithms *(3rd edition).*
- *For exercises 2 and 3, it is important that your solutions are presented in a readable form. In particular, you should provide precise descriptions of your algorithms using pseudo-code or reference existing pseudo-code from the textbook CLRS. To get partial points for not completely correct answers, it is also worth to write two or three lines in English to describe informally what the algorithm is supposed to do.*
- *Make an effort to use a readable handwriting and to present your solutions neatly.*

Exercise 1 [60 points in total]

Note that some of the following questions may have more than one correct solution. Mark the checkboxes for **ALL** correct solutions.

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

1.1. (*3 points*) $n \lg n^5 + n \lg 2^n + n\sqrt{n}$ is:

☐ **a)** $\Theta(n \lg n)$ ☐ **b)** $\Theta(n)$ ☐ **c)** $\Theta(\sqrt{n})$ ☐ **d)** $\Theta(n^2)$ ☐ **e)** $\Theta(n^{1.5})$

1.2. (*3 points*) $700 \cdot n^2 + 999 \cdot n^2 \lg n + 0.1 \cdot n^2 \lg^2 n$ is:

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

2. (*6 points*) Please choose the function(s) which make the following statement be true.

$$\lg(f(n) \cdot g(n)) = \Theta(n)$$

☐ **a)** $f(n) = n^2,\ g(n) = n^2$
☐ **b)** $f(n) = 2^n,\ g(n) = 2^n$
☐ **c)** $f(n) = 2^n,\ g(n) = 4^n$
☐ **d)** $f(n) = n \lg n,\ g(n) = 1$

3 (*8 points*) Consider the following recurrence:

$$\begin{aligned} T(1) &= \Theta(1) \\ T(n) &= 8 \cdot T(n/2) + n^4 \qquad (n > 1), \end{aligned}$$

Which of the following is correct? T(n) is:

☐ **a)** $\Theta(n^4 \lg n)$ ☐ **b)** $\Theta(n^2)$ ☐ **c)** $\Theta(n^3)$ ☐ **d)** $\Theta(n^4)$

4. (*8 points*) Given an array that is already sorted, e.g., $(1, 2, 3, 4, 5, 6, 7, 8)$. Which of the following algorithms will achieve their worst case?

- ☐ **a)** Insertion sort
- ☐ **b)** Selection sort
- ☐ **c)** Merge Sort
- ☐ **d)** Quick Sort

5. (*8 points*) Given array $A = (50, 17, 20, 30, 60, 55, 33, 27)$, check all the possible sequences which can trun A into a **max-heap**.

- ☐ **a)** Max-Heapify(A, 4), Max-Heapify(A, 2), Max-Heapify(A, 3), Max-Heapify(A, 1)
- ☐ **b)** Max-Heapify(A, 3), Max-Heapify(A, 4), Max-Heapify(A, 2), Max-Heapify(A, 1)
- ☐ **c)** Max-Heapify(A, 1), Max-Heapify(A, 2), Max-Heapify(A, 4), Max-Heapify(A, 3)
- ☐ **d)** Max-Heapify(A, 3), Max-Heapify(A, 4), Max-Heapify(A, 1), Max-Heapify(A, 2)

6. (*8 points*) Consider the following algorithm DoSomething(Array A). Note that indentations indicate block structures. Here, we assume that an array starts its index from 1 and $A.size()$ returns the number of elements in the array A.

```
DoSomething(Array A)
int n ← A.size();
Initialize an empty stack S with size 2 · n;
for x ← 1; x <= n; x++ do
    Push(S, A[x]);
for i ← 1; i <= n; i++ do
    Pop(S);
    int j ← 1;
    while !Stack-Empty(S) do
        A[j] ← Pop(S);
        j++;
    for k ← 1; k <= n − i; k++ do
        Push(S, A[k]);
```

Assuming the size of array A is n, What is the run time of DoSomething(A)?

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

7 (*8 points*) Consider a hash table with 9 slots using open addressing with quadratic probing to address conflicts. The hash function is

$$h(k, i) = (k + 3 \cdot i + i^2) \mod 9$$

After inserting 1, 5, 8, 10, 13, and then 17 into an empty hash table, what does the hash table look like now?

☐ **a)** `1, 10, 17, 5, 13, empty, empty, empty, 8`

☐ **b)** `empty, 10, 1, 17, 13, empty, 5, empty, 8`

☐ **c)** `empty, 1, 10, 13, empty, 17, empty, 5, 8`

☐ **d)** None of the above is correct. Write the correct one here: ___________

8 (*8 points*) Suppose that we are searching for number 15 in a **binary search tree**. Which **one** of the following four sequences could be a sequence of tree nodes examined in the search? (Note that 15 may or may not be in the tree.)

☐ **a)** `100, 23, 7, 8`

☐ **b)** `21, 7, 18, 5, 3, 11`

☐ **c)** `9, 11, 16, 18, 13, 20`

☐ **d)** `8, 12, 23, 14, 21, 19`

Exercise 2 [20 points]

A student participates in an exam which includes N different questions $\{Q_1, Q_2, \ldots, Q_N\}$. Each question Q_i (where $1 \leq i \leq N$) has a value in points, denoted by v_i, and also the time (in minutes) to solve the question, denoted as t_i. For example, consider the following exam with 5 questions.

	Q_1	Q_2	Q_3	Q_4	Q_5
Value v_i (points)	20	10	5	2	1
Time to complete t_i (minutes)	10	6	4	2	1

For the first question Q_1, solving it requires 10 minutes and the student can get 20 points.

Now, given a limited time, say X minutes, which is often not enough to solve all questions, we try to identify the questions which the student need to solve such that the student can get the highest points.

For example, if $X = 15$, the student can solve $\{Q_1, Q_3\}$, which take 14 mins, and the student gets 25 points; the student can also solve $\{Q_1, Q_4, Q_5\}$, which take 13 mins, and the student gets 23 points. Comparing this two options, solving $\{Q_1, Q_3\}$ is a better option because the two questions Q_1 and Q_3 can be solved within the limited time 15 mins and give higher points.

1. (*5 points*) Consider the following naive algorithm to solve the problem: (1) it enumerates all possible question combinations; (2) for each question combination, it computes the sum of points and the total required time; (3) it returns the combination with the highest sum of points that is within the time limit X. What is the asymptotic complexity of the naive algorithm? For the specific example shown in the above table, how many question combinations the naive algorithm needs to enumerate?

2. (*15 points*) Design a *dynamic programming* algorithm to solve the problem. In your design, please (1) briefly write down your ideas and write down the recurrence; (2) write down pseudo code; (3) identify the asymptotic complexity of your algorithm.

Exercise 3 [20 points]

A Kommune decides to build n new villages $v_1, v_2, \ldots, v_n$ and provide electricity to all villages. The kommune has two options:

1. Build a power station at village v_i, which can provide unlimited electric power. The cost for building a power station for v_i is c_i.

2. Build a transmission connection between two different villages v_i and v_j. If one village has a power station already, the electricity can go to the other village. The cost for building a transmission connection between v_i and v_j is t_{ij}.

Consider a specific example where we have 4 new villages v_1, v_2, v_3, v_4. The costs of the two options are shown in Tables 1 and 2.

	v_1	v_2	v_3	v_4
Cost c_i	5	4	4	3

Table 1: Power Station Building Costs according to Option 1

	v_1	v_2	v_3	v_4
v_1	0	2	2	2
v_2	2	0	3	3
v_3	2	3	0	4
v_4	2	3	4	0

Table 2: Transmission Connection Building Costs according to Option 2

For example, building a new power station at village v_1 costs 5; and building a transmission connection between villages v_1 and v_2 costs 2.

Given a set of n new villages, the kommune would like to design a solution to minimize the cost of providing electricity to all villages.

1. (*6 points*) Given the 4 villages and the costs shown in Tables 1 and 2, write down (1) the minimum cost (i.e., a single value) of providing electricity to the 4 villages; and (2) how do you get this value—which villages are build with new power stations and which village pairs are connected by transmission connections. (*Hint: in order to provide electricity, you have to build at least one power station. Thus, it is often a good idea to build a power station at the village with the cheapest power station building cost.*)

2. (*6 points*) We would like to solve the problem as a graph problem. Show how to model this problem as a graph problem. Specifically, write down what do vertices and edges represent. If weighted graphs are used, write down what do weights represent. You need to model the problem in a general manner, e.g., a graph that models n new villages but not just 4 villages shown in the example. (*Hint: Try to extend the classic minimum span tree problem.*)

3. (*8 points*) Design a graph algorithm to solve the problem. Analyze the worst-case running time complexity of your algorithm. (*Hint: Try to modify the Prim's algorithm.*)