Algorithms and Data Structures (DAT3/SW3)

Exam Assignments

Chenjuan Guo & Bin Yang

10 January 2020

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

This exam consists of three exercises and there are **two** hours (10.00 a.m. to 12.00 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 an old fashioned calculator.

- *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 each following question has only a single correct solution.

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

1.1. (*5 points*) $9^{\log_3 n} + 0.003 \cdot \lg^n 27 + 77892 \cdot n^{9.2} + 57lgn^5$ is:

☐ **a)** $\Theta((lg27)^n)$ ☐ **b)** $\Theta(n)$ ☐ **c)** $\Theta(lgn^5)$ ☐ **d)** $\Theta(n^2)$

1.2. (*5 points*) $(\lg n) \cdot (5 \cdot lgn + 0.366 \cdot n^2 + 90000 \cdot n - 288)$ is:

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

2 (*10 points*) Consider the following recurrence:

$$\begin{aligned} T(1) &= 60 \\ T(n) &= T(3n/4) + \sqrt{n} \qquad (n > 1), \end{aligned}$$

2.1. (*2 points*) Is this recurrence in the format that can be solved by the master method? If yes, choose which case should be used?

☐ **a)** No, the master method cannot solve the recurrence
☐ **b)** Yes, case 1 should be used
☐ **c)** Yes, case 2 should be used
☐ **d)** Yes, case 3 should be used

2.2. (*4 points*) If you could apply the master method, choose possible a, b, and ϵ values.

☐ **a)** $a = 1, b = \frac{3}{4}, \epsilon = 0.5$ ☐ **b)** $a = 1, b = \frac{4}{3}, \epsilon = 0.5$
☐ **c)** $a = 1, b = \frac{3}{4}, \epsilon = 0$ ☐ **d)** $a = 1, b = \frac{4}{3}, \epsilon = 0$

2.3. (*4 points*) Choose the correct solution for $T(n)$.

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

3. (*10 points*) Given an array $[26, 54, 18, 2, 9, 17, 45]$, consider selection sort, merge sort, quick sort, and insertion sort.

For quick sort, the element at the first position of a (sub)array is chosen as a pivot. For example, 26 is chosen as the pivot in the beginning.

3.1. (*4 points*) Which of the following statements is true?

☐ **a)** The worst case time complexity of merge sort is $\Theta(n^2)$.
☐ **b)** The average case time complexity of quick sort is $\Theta(n \lg n)$.
☐ **c)** The best case time complexity of selection sort is $\Theta(n \lg n)$
☐ **d)** Quick sort is not an in place sorting algorithm.

3.2. (*6 points*) $[2, 9, 17, 26, 54, 18, 45]$ comes from applying 3 steps of which sorting algorithm? Specifically, for a recursive sorting algorithm, applying 3 steps means to call the sorting algorithm 3 times; for a non-recursive sorting algorithm, applying 3 steps means to run the outer loop 3 times in the sorting algorithm.

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

4. (*10 points*) We implement a priority queue using a min-heap.

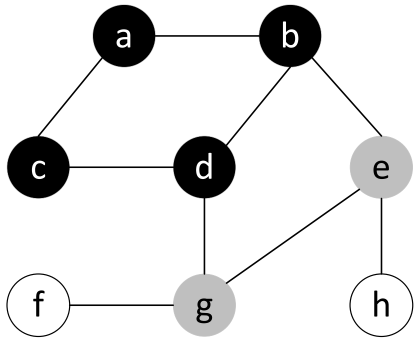
4.1. (*5 points*) Given an array $A = [26, 54, 18, 2, 9, 17, 45]$, which of the following arrays is the output A' from running BUILD-HEAP(A) for min-heap.

☐ **a)** $A' = [26, 9, 18, 2, 9, 54, 45]$ ☐ **b)** $A' = [2, 9, 17, 18, 26, 45, 54]$
☐ **c)** $A' = [26, 2, 17, 54, 9, 18, 45]$ ☐ **d)** $A' = [2, 9, 17, 54, 26, 18, 45]$

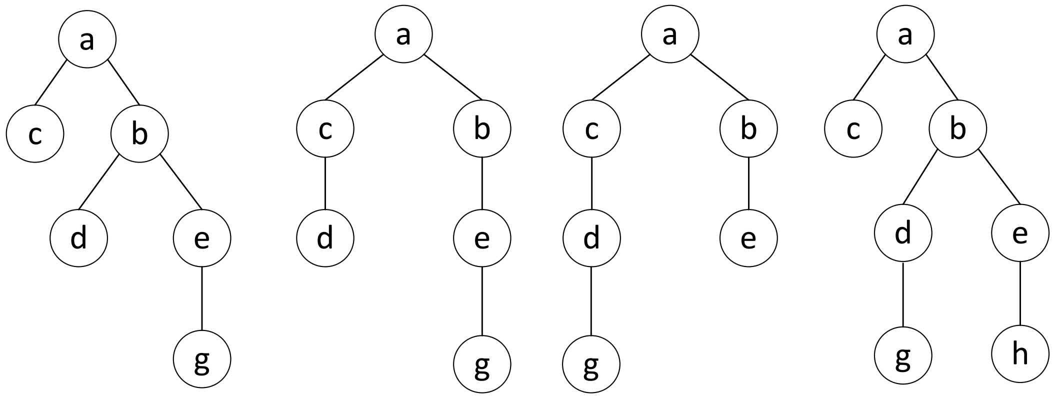
4.2. (*5 points*) Currently, the priority queue implemented by the min-heap holds the output A' obtained from **4.1.** After we run EXTRACT-MIN(A') on the priority queue, which of the following arrays is correct?

☐ **a)** $[45, 9, 17, 54, 26, 18, 2]$ ☐ **b)** $[9, 45, 17, 54, 26, 18, 45]$
☐ **c)** $[9, 26, 17, 54, 45, 18, 45]$ ☐ **d)** $[9, 17, 18, 26, 45, 54, 2]$

5. (*10 points*) Given a graph as follow, we use *a* as the source vertex and run a breath first search (BFS) to build a breath first tree.



5.1. (*7 points*) During BFS, vertices are marked with different colors, i.e., white, gray, and black, to track progress (c.f. CLRS. Page 594 or slides). According to the color labeled for each vertex, choose a correct breath first tree that is partially built.



☐ a) ☐ b) ☐ c) ☐ d)

5.2. (*3 points*) According to the vertex color and the breath first tree built as the result of **5.1.**, which vertex is the next one that will be put into the breath first tree?

☐ **a)** *h* ☐ **b)** *f* ☐ **c)** *g* ☐ **d)** *e*

6. (*10 points*) Consider a directed graph $G = (V, E)$, where $V = \{a, b, c, d, e, f, g, h\}$. Its adjacency-matrix representation is given as follows. Let's run a topological sorting algorithm on the graph G.

	a	b	c	d	e	f	g	h
a	0	1	0	0	0	1	0	1
b	0	0	0	0	1	0	0	0
c	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0
e	0	0	1	0	0	0	0	0
f	0	0	0	0	1	0	0	0
g	0	0	0	1	0	1	0	0
h	0	0	0	0	0	1	0	0

6.1. (*3 points*) Which algorithm does the topological sorting algorithm need?

☐ **a)** Prim's Algorithm ☐ **b)** Breath First Search Algorithm
☐ **c)** Depth First Search Algorithm ☐ **d)** Dijkstra's Algorithm

6.2. (*7 points*) Which of the following choices are possible sequences of topologically sorted vertices?

☐ **a)** `g d a f h b e c`
☐ **b)** `b a h g f e c d`
☐ **c)** `g d h a b f e c`
☐ **d)** `a b g d h f e c`

Exercise 2 [20 points]

You are traveling by ferries down a river and there are n stations along the way. Before starting your journey, you are given the fee f_{ij} in order to travel from station i to station j, with $1 \leqslant i < j \leqslant n$. These fees are arbitrary. Be aware that you are not allowed to travel back, i.e., from station j to station i, with $1 \leqslant i < j \leqslant n$. An example for travel fees is given as follows. For example, the travel fee from station 1 to station 3 is $f_{13} = 2$, from station 3 to station 5 is $f_{35} = 5$, and from station 1 to station 5 is $f_{15} = 8$.

	1	2	3	4	5
1	0	4	2	1	8
2	-	0	9	7	4
3	-	-	0	4	5
4	-	-	-	0	6
5	-	-	-	-	0

You must begin at station 1 and end at station n, and you are allowed to stop in middle stations, i.e., stations 2 to $n - 1$. For example, you begin at station 1 and end at station 5. If you stop at station 3, the fee from station 1 to station 5 is $f_{13} + f_{35} = 7$. You could also travel directly from station 1 to station 5, then the fee is $f_{15} = 8$. You may think of other ways of traveling, and your goal is to minimize the total travel fee from station 1 to station n.

Let's define $m[i]$ as the minimum travel fee from station 1 to station i, with $1 \leqslant i \leqslant n$.

1. (*7 points*) Write down the recurrence for $m[i]$.
   (Hint: how could you recursively define $m[i]$? What about $m[1]$? What about $m[i]$ when $i > 1$?).

2. (*7 points*) Design a dynamic programming algorithm to solve the problem. Write down the pseudo code.

3. (*6 points*) Analyze the time complexity of your algorithm: write down the time complex and analyze the process.

Exercise 3 [20 points]

Let's consider a route planning problem. Assume we have a small road network that has 5 cities: a, b, c, d, and e. There exist roads connecting from a to b with 2 km, from a to c with 13 km, from b to d with 3 km, from b to e with 5 km, from d to c with 4 km, from e to c with 4 km, from d to a with 2 km, and from d to e with 1 km. Further, the road from e to c are downhill, and all the other roads are uphill or flat. Note that all these roads are **one-way** roads.

Assume that we have two cars, one diesel car and one electric car. For the diesel car, the diesel consumption is 1 unit per km, no matter traversing on a flat, an uphill or a downhill road. For the electric car, it consumes electrical energy when traversing on a uphill or flat road, and the electrical energy consumption is 1 unit per km. In contrast, the electric car recharges itself when traversing on a downhill road, and it gets 1.5 unit electrical energy by traversing 1 km downhill road.

Given a source city and a destination city, the route planning problem needs to find the route that uses the least amount of diesel from the source city to the destination city for a diesel car, and to find the route that uses the least electrical energy from the source city to the destination city for an electric car.

Choose a question set and write down its solutions. DO NOT write solutions for both question sets. REMEMBER to write down which question set you choose.

If you choose question set A, you can only work on it and your solutions to question set B would not count into the final grade, and vice versa.

Choose Question Set:

Question Set A (Diesel Cars):

1. (*4 points*) Draw the corresponding graph that models the route planning problem for diesel cars. Write down the **adjacency list representation** of the graph.

2. (*16 points*) Given source city a and destination city c, run an algorithm to identify the route that uses the least amount of diesel on the graph that you just drew.

   (a) (*4 points*) Pick up an algorithm from the 3 algorithms that we have learned for solving the route planning problem. Write down the name of the algorithm that can solve the route planning problem for diesel cars, satisfying that it is the most efficient and feasible algorithm. Explain briefly about why you choose the algorithm.

(b) (*6 points*) Write down the output route and the amount of diesel used.

(c) (*6 points*) Write down the number of times that **edge relaxations** are executed and **have improved** the existing distance from the source to a vertex in the algorithm. If you do not remember the edge relaxation operation, check P. 648-649, CLRS.

Question Set B (Electric Cars):

1. (*4 points*) Draw the corresponding graph that models the route planning problem for electric cars. Write down the **adjacency matrix representation** of the graph.

2. (*16 points*) Given source city a and destination city c, run an algorithm to identify the route that uses the least amount of electrical energy on the graph that you just drew.

   (a) (*4 points*) Pick up an algorithm from the 3 algorithms that we have learned for solving the route planning problem. Write down the name of the algorithm that can solve the route planning problem for electric cars, satisfying that it is the most efficient and feasible algorithm. Explain briefly about why you choose the algorithm.

   (b) (*6 points*) Write down the output route and the amount of electrical energy used.

   (c) (*6 points*) Write down the number of times that **edge relaxations** are executed in the algorithm. If you do not remember the edge relaxation operation, check P. 648-649, CLRS.