Advanced Algorithms

(DAT6/SW6/IT8)

Exam Assignments

Bin Yang

10.00 - 12.00, 4 June 2019

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

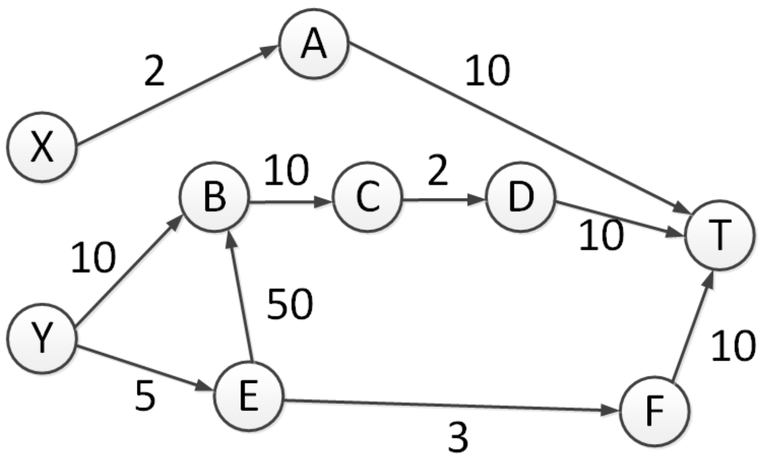
This exam consists of two exercises. Exercise 1 is a set of quizzes. Exercise 2 has a few open questions. When answering the quizzes in Exercise 1, mark the check-boxes or write down numbers, matrices, or sentences on these papers. When answering the questions in Exercise 2, remember to put your name and your student number on all additional sheets of papers that you will need to use.

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

- *Read* carefully *the text of each exercise before solving it! Pay particular attentions to the terms in* **bold***.*
- *For Exercise 2, it is important that your solutions are presented in a readable form. Make an effort to use a readable handwriting and to present your solutions neatly.*

Exercise 1 [50 points in total]

1. We run Edmonds-Karp algorithm on the following flow network. Note that there are two sources, vertex X and vertex Y, and one sink, vertex T.



(*9 points*) Write down all the critical edges when Edmonds-Karp algorithm is executed.

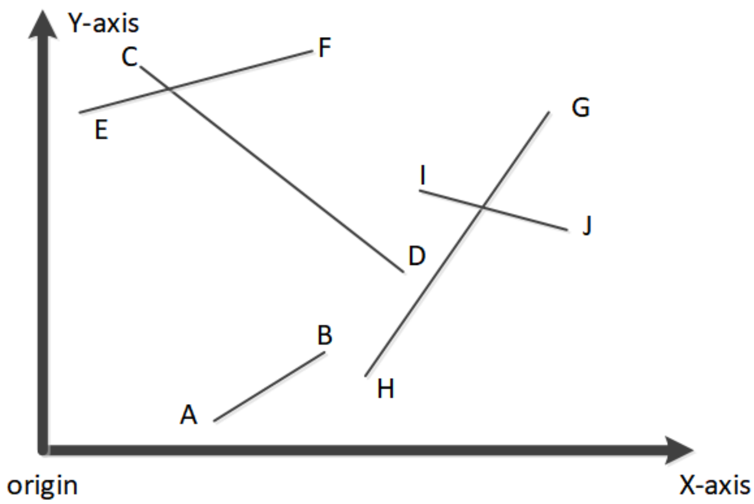
(*4 points*) Write down the identified maximum flow: ________________

2. (*12 points*) Consider a document with the following characters along with their frequencies.

Character	A	B	C	D	E	F	G
Frequency	12	9	7	15	5	13	1

Assume that we use Huffman coding to encode the characters, write down the code for "A": ________________ and "D": ________________

3. (*13 points*) Consider the following 5 line segments shown in the following figure.



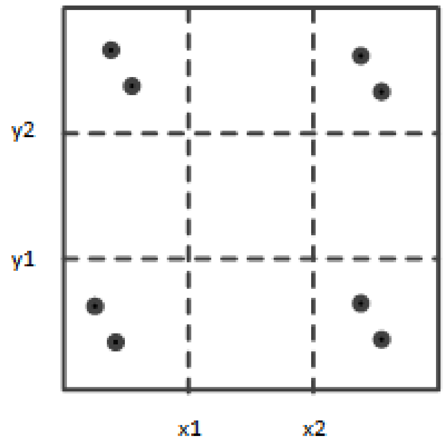
Let's use the efficient sweeping technique from Lecture 6 (i.e., the algorithm with $\Theta(n \log n)$ runtime where n is the number of line segments) to check if there are line segments that intersect. But instead of using a vertical sweeping line from left to right, we use a horizontal sweeping line from bottom to top.

The idea is similar. For each line segment, we have a low end point and a high end point. When we see a low end point, we insert the corresponding line segment into the sweep-line status; and when we see a high end point, we remove the corresponding line segment from the sweep-line status.

Based on the above algorithm, at which end point, we identify that there exists two line segments that intersect each other? Please write down the end point here:

____

4. (*12 points*) We have some points spread over a rectangular region. Assume that the region is partitioned into 9 sub-regions according to two vertical lines and two horizontal lines (see the dashed lines shown in the following figure).



Assume that there exists one fourth data points in the upper left sub-region, upper right sub-region, lower left sub-region, and lower righ sub-region, respectively. If we want to process a range query $[x_1, x_2], [y_1, y_2]$, which indexing structure gives the best query run time? We assume that the indexing structures have already been built in advance.

☐ **a)** Use two 1D BSTs.
☐ **b)** Use a 2D Range Tree.
☐ **c)** Use a KD-tree.
☐ **d)** They have the same asymptotic query time complexity.

Exercise 2 [50 points in total]

1 (*25 points*) In Lecture 5, we have studied amortized analysis. Now let's consider a sequence of n operations. Assume that the cost of the i-th operation satisfies the following:

1. If i is a power of 3, the cost of the i-th operation is i.

2. If i is not a power of 3, the cost of the i-th operation is 1.

For example, the following table shows the cost for each operation in the first 10 operations.

i-th Operation	1	2	3	4	5	6	7	8	9	10
Cost	1	1	3	1	1	1	1	1	9	1

Identify the amortized cost **per operation**. Please be as specific as possible. For example, if the amortized cost is a constant, try to identify the constant, e.g., 2 or 3, rather than saying $\Theta(1)$.
Hint: You may need to use the following equation.

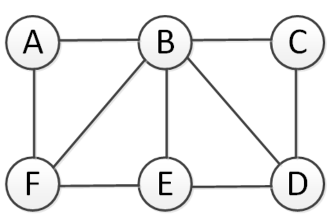
$$\sum_{k=0}^{n} x^k = \frac{x^{n+1} - 1}{x - 1}$$

2 Let's first recall what is a **Hamiltonian cycle** in an undirected graph. Given an undirected graph $G = (V, E)$, a path $(v_0, v_1, v_2, \ldots, v_k)$ forms a **cycle** if $k \geq 3$ and $v_0 = v_k$; the cycle is **simple** if $v_1, v_2, \ldots, v_k$ are distinct. A Hamiltonian cycle is a **simple cycle** that contains each vertex in V.

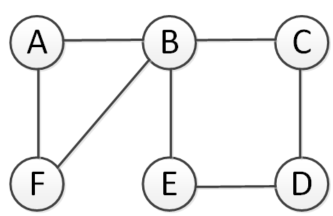
The Hamiltonian Cycle problem is a decision problem that determines whether or not a given undirected graph contains a Hamiltonian Cycle. If it contains, then also print the Hamiltonian Cycle.

- (*11 points*) Design a backtracking algorithm to solve the Hamiltonian Cycle problem. Define what is a configuration, i.e., what is the remaining subproblem to be solved and what is the set of choices made so far. What kind of configuration corresponds to a deadend and what kind of configuration corresponds to a solution.

- (*14 points*) Using the designed backtracking algorithm to solve the Hamiltonian Cycle problem on the following two undirected graphs. Assume that the algorithm always starts from vertex A. Draw the search spaces and mark deadends and solutions (if solutions exist) for both graphs.



Graph 1



Graph 2