1.	
2.	
3.	
Sum	

Re-exam

Algorithms and Data Structures (Dat 1 and SW 3)

19.02.09

Finn V. Jensen

Full name	
CPR number	
e-mail at cs.auc.dk	

Read first

- Start by filling in your name, CPR number, and e-mail address. Remember to put your name and CPR number also on any additional sheet of paper you will use.
- The exam consists of three questions. All questions are relative to the topics covered during the lectures and to the content of the main textbook by Thomas H. Corman et al. You have 3 hours to answer the questions. It is recommended to spend approximately one hour per question. Make sure that your writing font is readable.
  - In Question 1 mark the correct answer out of the offered choices, or write the result directly into the highlighted box. For any calculations you need to do use a separate sheet of paper (this paper is not a part of the answer, do not hand it in). Do not provide any additional justification of your answers in this question.
  - In Questions 2 and 3 read carefully what you are asked to do. Make sure that you answer all the subquestions, and only those. Your answers should be clear and precise. Whenever you are asked to write an algorithm, use your favorite pseudo-code (the one used during the lectures is recommended but not required). It is also worth to write two or three lines describing informally what the algorithm is supposed to do.
- It is not allowed to use any electronic devices like laptops and mobile phones, except for a pocket calculator (though it will not be needed). You can freely use your notes from the lectures, textbooks, or any other course material.

Question 1 (*In total 50 points*)

a) (*6 points*)
Mark the correct answer by putting a cross in the corresponding field.

$\frac{n^3+2n^2-5n+7}{2n+3} + 10n + 117$ is	☐ $\Theta(n^3)$	☐ $\Theta(n^2)$	☐ $\Theta(n)$
$\lg(n^2 + 2^n) + 12n^2$ is	☐ $\Theta(n^2)$	☐ $\Theta(n \lg n)$	☐ $\Theta(2^n)$
$2^{2\lg n} + n^{1.5} + n(\lg n)^2$ is	☐ $\Theta(n(\lg n)^2)$	☐ $\Theta(n^2)$	☐ $\Theta(n^{1.5})$

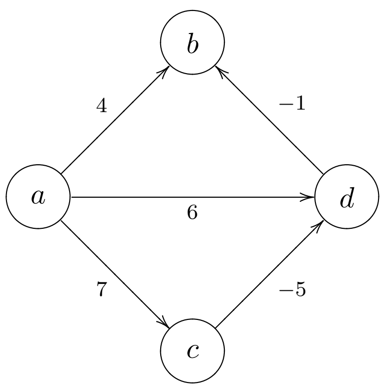
b) (*8 points*)
Consider the priority queue below, organized as a max-heap.

1	2	3	4	5	6	7
7	6	4	3	2	1	

Write in the box below the result of first inserting "5" (MAX-HEAP-INSERT) and then calling HEAP-EXTRACT-MAX.

1	2	3	4	5	6	7

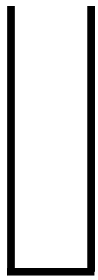
c) (*7 points*)
Fill in the table for computing shortest paths using the Bellman-Ford algorithm. The edges are taken in the order $(a, b), (a, c), (a, d), (d, b), (c, d)$



	init	$i = 1$	$i = 2$	$i = 3$
a	0			
b	∞			
c	∞			
d	∞			

d) (*7 points*)
Fill in the stack s below with values resulting from the following operations.

s:STACK; q:QUEUE; s.make; q.make
q.enqueue(1)
q.enqueue(2)
q.enqueue(3)
s.push(q.dequeue)
s.push(4)
q.enqueue(s.pop)
s.pop
s.push(q.dequeue)
q.enqueue(5)
s.push(q.dequeue)
s.push(q.dequeue)



e) (*7 points*)
Given a hash function $h(k) = (5 \cdot k)\ mod\ 7$, and a key sequence 1, 11, 8, 17, 2, 15. Insert all keys into the hash table $A[0..6]$ below. Use linear probing to resolve collisions.

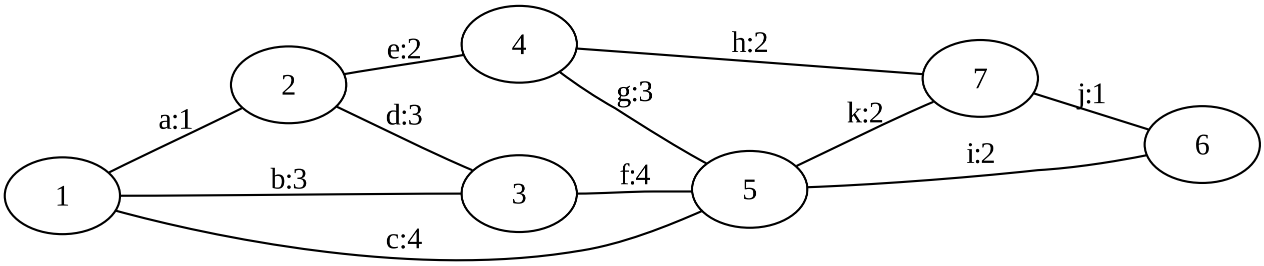
0	1	2	3	4	5	6

f) (*8 points*)
Consider the following recurrence equation (for simplicity, assume that n is a power of 2):

$T(1) = 1$
$T(n) = 8 \cdot T(\frac{n}{2}) + n^2 \quad$ for $n \geq 2$.

$T(n)$ is
☐ $\Theta(n^2)$ ☐ $\Theta(n^2 \lg n)$ ☐ $\Theta(n^3)$ ☐ $\Theta(n^3 \lg n)$

g) (*7 points*)



From the set of edges $a, b, c, d, e, f, g, h, i, j, k$ list all edges which cannot be part of a maximal spanning tree for the graph above (Note: there may be too much space in the box below).

, , , , , , ,

Question 2 (*25 points*)

This exercise concerns transport. Consider the graph below. The labels on the edges are capacities.

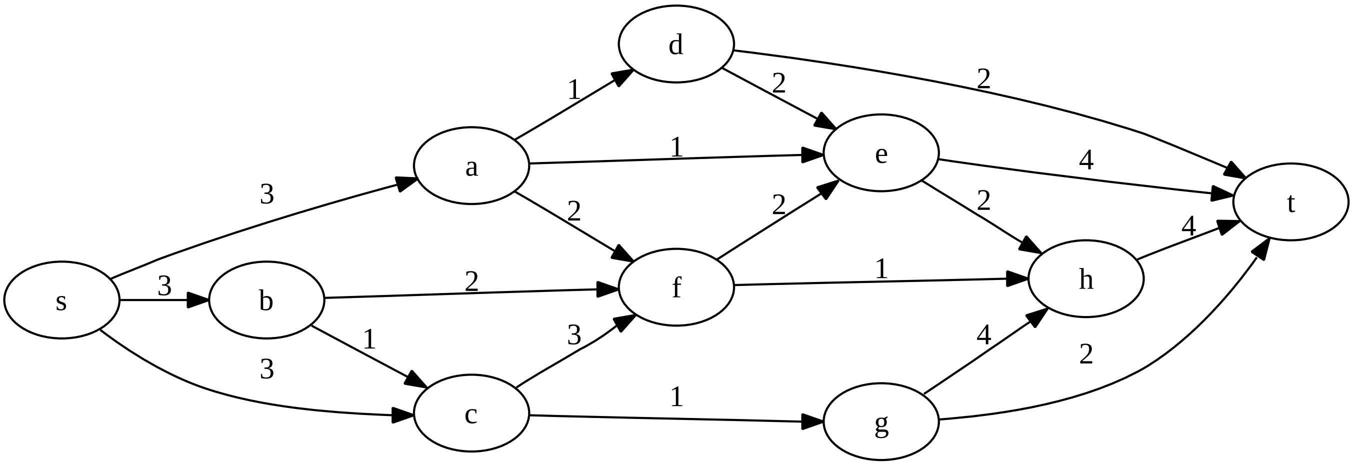


Figure 1: Transport graph for Question 2

i) Calculate the maximal flow from s to t and draw your final residual network.

ii) You wish to increase the maximal flow from s to t by 1. You will do this by increasing the capacity of one edge. Use your residual network to give a complete list of relevant edges.

iii) Design an algorithm for the following problem: Given a directed graph with capacities on the edges and with a source node s and sink node t. Given an edge $X \to Y$. Determine whether an increase of the capacity of $X \to Y$ will increase the maximal flow from s to t. What is the time complexity of your algorithm?

Question 3 *(25 points)*

This question is inspired by a scientific problem concerning establishing optimal troubleshooting sequences for car engines.

You work with a decreasing sequence of real positive numbers $(x_1, \ldots, x_n)$. The sequence can be *split*: $(x_1, \ldots, x_i) \bigodot (x_{i+1}, \ldots, x_n)$, and the two subsequences can recursively be split again. The results are called *split expressions*. For example, if you have the sequence (a, b, c), then you can build the following four split expressions: $(a, b, c), (a, b) \bigodot (c), (a) \bigodot (b, c), (a) \bigodot (b) \bigodot (c)$.

a) Prove that a sequence of n numbers has 2^{n-1} different split expressions.

There is a cost function Γ over split expressions:

i) $\Gamma((y_1, \ldots, y_n)) = y_1(n + 2)$
ii)$\Gamma(X \bigodot Y) = \Gamma(X) + \Gamma(Y)$, where X and Y are split expressions.

b) Calculate $\Gamma((4, 2, 1) \bigodot (0.5))$ and determine for the sequence $(4, 2, 1, 0.5)$ a split expression of minimal cost.

c) Design an $O(n^3)$ algorithm for determining a split expression of minimal cost for a sequence with n numbers.