1.	
2.	
3.	
Sum	

Exam

Algorithms and Data Structures (Dat 1 and SW 3)

30.1.2007

Finn V. Jensen

Full name	
CPR number	
e-mail at cs.aau.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. Cormen 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 device like laptop or mobile phone, except for a basic calculator (though it will not be needed). You can freely use your notes from the lectures, textbooks, course material from the folder, and copies of slides.

Question 1 (*In total 50 points*)

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

$\frac{n^6}{1021} + 7n^2 + 60n$ is

☐ $\Theta(n \log n)$ ☐ $\Theta(n^2)$ ☐ $\Theta(n^3)$ ☐ $\Theta(n^6)$

$3n^3 + 5n^2$ is

☐ $\Theta(n^n)$ ☐ $\Theta(n^2)$ ☐ $\Theta(n^3)$ ☐ $\Theta(n^6)$

$\log 3n^n + 5$ is

☐ $\Theta(n^n)$ ☐ $\Theta(n \log n)$ ☐ $\Theta(n^3)$ ☐ $\Theta(n^6)$

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

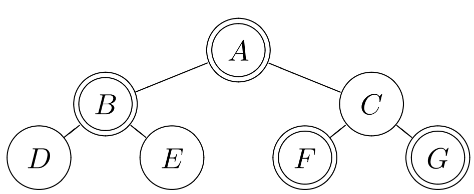
Consider the recurrence (for simplicity, assume that n is a power of 3).

$T(1) = 1$
$T(n) = 4T(\frac{n}{3}) + \frac{1}{2}n^2$.

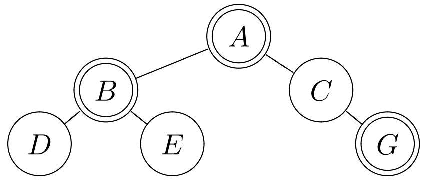
$T(n)$ is

☐ $\Theta(n^3)$ ☐ $\Theta(n^{log_3 4})$ ☐ $\Theta(n^{log_3 4} \log\, n)$ ☐ $\Theta(n^2)$

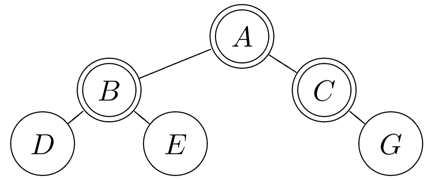
c) (*8 points*)
Below you see a red-black tree (with NIL-leafs omitted; black nodes are double-circled).



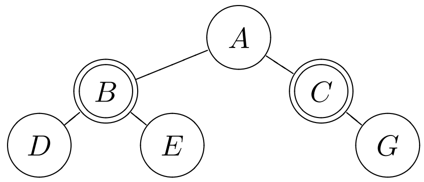
Which of the four trees below may be the result of removing F from the tree and re-establishing it as a red-black tree? (There may be more than one correct answer).



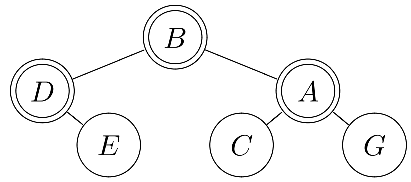
a)



b)



c)



d)

☐ *a)* ☐ *b)* ☐ *c)* ☐ *d)*

d) (*7 points*)

What is the time complexity of the algorithm? Mark all correct answers (there may be more than one).

```
x:=1
for i:= 1 to n do
   for j:= 1 to i do
      for k:= 1 to 133 do
         x=x*2
      end for
   end for
end for
```

☐ $O(n)$ ☐ $O(n \log n)$ ☐ $O(n^2)$ ☐ $O(n^3)$

e) (*7 points*)

Arrange the numbers {7, 6, 13, 8, 9, 2, 5, 11} such that they form a max-heap. Write the result in the array below.

1	2	3	4	5	6	7	8

f) (*7 points*)

The table below is a hash table for which the keys were stored using the quadratic hash function

$h(k, i) = (k + i + 3i^2) \bmod 11.$

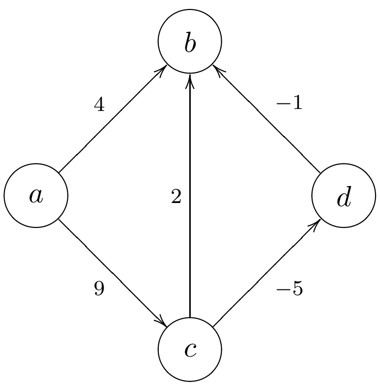
0	1	2	3	4	5	6	7	8	9	10
22		*88*	*17*	*4*	*6*	*28*	*26*	*15*	*31*	*10*

When looking up, for example, the key *15*, first the entry 4 is inspected, and as the key is not placed here, the entry 8 is inspected.
Write the series of entries inspected when looking up the key *26* (there may be less than 5 entries inspected).

1st	2nd	3rd	4th	5th

g) (*9 points*)

Fill in the table for computing shortest paths using Bellman-Ford algorithm.



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

Question 2 (*25 points*)

Consider an $n \times n$ board with positions (i, j) with $(i, j) \in \{1, ..., n\}$. The positions are referred to with (row, column). At each position (i, j) there is placed some money, $m(i, j)$ (non-negative integers). In the figure below you see an example 4×4 board.

1	5	7	1
4	0	3	7
5	7	5	8
3	5	9	1

If you are placed in position (i, j) you can move one step to the right, or one step down, or one step down-right-diagonal (provided that you do not cross the border of the board). That is, from position (i, j) you may move to $(i, j+1)$, $(i+1, j)$, or $(i+1, j+1)$. You can start in any position, perform legal moves, and collect the money at each position visited.

The path [(2, 1), (2, 2), (3, 2), (3, 3)] on the board in the figure collects 4+0+7+5 = 16. A path collecting a maximal amount is called a *maximal path*. A maximal path on the board in the figure is [(1, 1), (1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (4, 4)].

a) Find another maximal path on the board in the figure.

b) Prove that for any $n \times n$ board there is at least one maximal path starting in (1, 1) and ending in (n, n).

c) Prove that for any $n \times n$ board there exists a maximal path without moves of the diagonal type ((i, j) to $(i + 1, j + 1)$).

d) Construct an $O(n^2)$ algorithm for determining a maximal path.

Question 3 (*25 points*)

In this question you will be solving the **Pedigree Counting Problem**.

Let us consider a normal human family. Each person has two parents, one mother and one father. The mother and father also have two parents (grandmother and grandfather) and so on. For a given person, we can build a *complete* history of his/her ancestors up to a given number of generations. We shall call it a pedigree.
For example, for Peter the complete pedigree up to 3 generations may look as follows: Peter has a mother called Anna and a father called John; Anna has a mother called Rikke and a father Jan; and John has a mother Joan and a father Fredrik.
For simplicity we assume that for each person we remember only his/her first name. We also assume that pedigrees are *healthy*. A pedigree is healthy for a person P if each of P's ancestors is related to P in a unique way. This for example rules out the situation that the father of Peter's father (farfar) and the father of Peter's mother (morfar) is the same person.

a) Suggest a mathematical formalization of a pedigree for a person P. What data structure can be used to represent the pedigree? How do you represent people and their first names, how do you represent maternal and paternal relationships? What is the role of P in the abstract data type?

b) Show how Peter's pedigree above can be modeled in the data type you suggested (a drawing is sufficient).

c) In Peter's pedigree of 3 generations, there are in total 7 different persons. How many persons are involved in a (complete) pedigree of 4 generations and how many in a pedigree of n generations?

d) In a pedigree, it is very well possible that a person has several ancestors with the same first name (though they are different individuals). Write an algorithm which for a given pedigree counts how many persons in the pedigree are called Kim. What is the time complexity of the algorithm (Θ-notation)?

e) (**extra 10 points**) Consider now pedigrees that are not necessarily healthy. What data structure can be used to model such pedigrees? Given a pedigree represented by your data structure, how can you algorithmically decide whether it is healthy or not? Provide an algorithm to count how many persons in a given (possibly unhealthy) pedigree are called Kim.