M.Sc. Qualification exam – 2014

Date: 21.03.14
Time: 3 hours
Answer **4 out of 6 questions**

Important: In the following table, circle the numbers of the four questions you chose to answer.

Question	Grade
1	/25
2	/25
3	/25
4	/25
5	/25
6	/25

Question 1: Data structures

A. Consider a website that people can upload funny videos to and other people can watch. Assume that each video is associated with a name and length (viewing time in minutes). You can assume that the names and the lengths of each video are unique. Assume you are given n videos. The website should support the following operations:

   i. Add a new video.
   ii. Find the shortest video.

   a. Suggest an efficient data structure (with respect to **worst-case** time complexity) that allows to perform these operations. Give a short explanation.
   b. What is the time complexity for inserting the n videos into the data structure? Give a short explanation.

B. Assume the website decides to improve its service and allow an additional operation:

   iii. Output five names of videos, if exists, such that each has length longer than t_{min} and shorter than t_{max}.

   a. Suggest an efficient data structure (with respect to **worst-case** time complexity) that allows performing the three operations. Give a short explanation.
   b. What is the time complexity for inserting the n videos into the data structure? Give a short explanation.

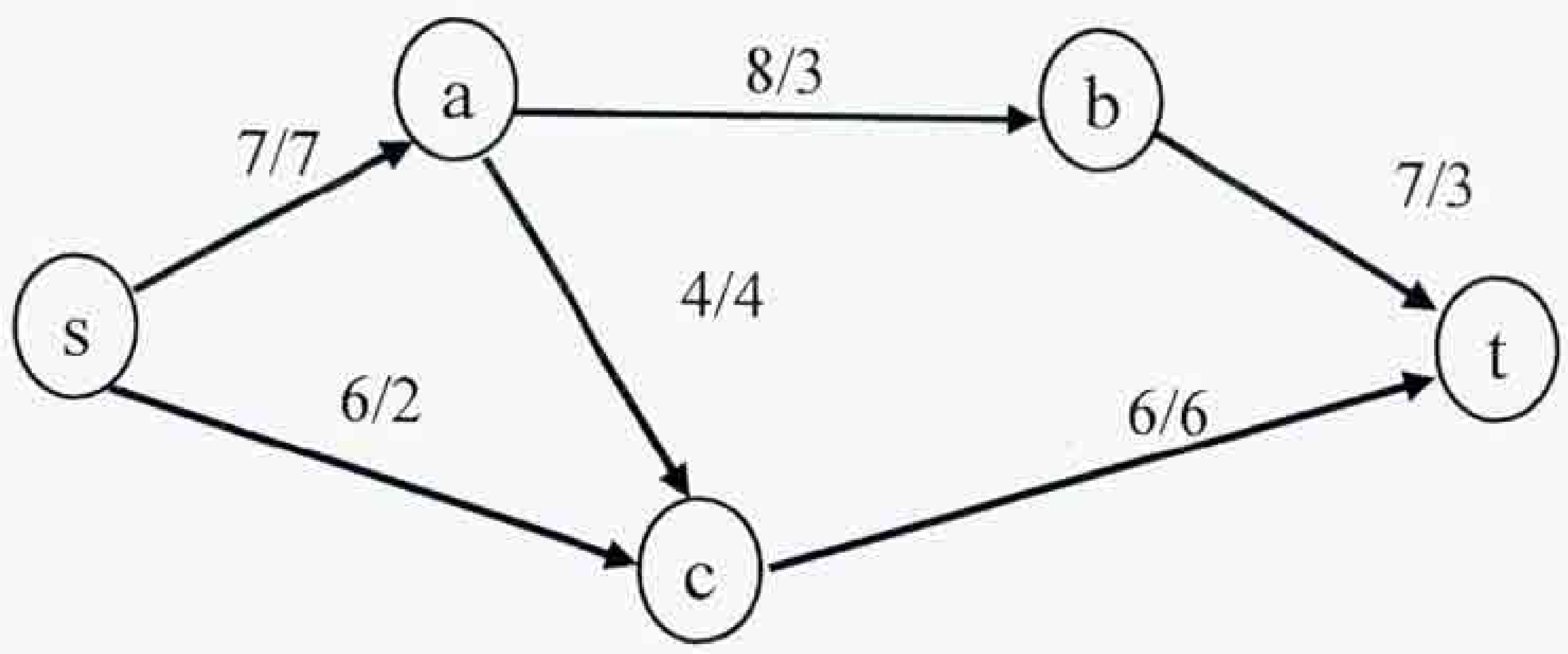
C. (This part is independent of parts A and B). The worst-case time complexity of a given algorithm is $\theta(n^3 \log n)$. Write for each of the following claims whether they are necessarily true, necessarily false, or may be true. Give a short explanation.

  a. There exists an input of length 10 for which the algorithm runs 10^4 steps.
  b. The worst time complexity of the algorithm is also $O(n^4)$.
  c. For each $n > 10$ there exists a run of the algorithm that takes $4n^3$.
  d. For each $n < 10$ there exists a run of the algorithm that takes $4n^5$.

Question 2: Algorithms

1. (15 pts.) Let $G=(V,E)$ be a simple undirected connected graph, with positive weights on the edges.
Describe an algorithm whose time complexity is $O(|E| \log |E|)$ that returns a set of edges $E' \subseteq E$ that should be removed from the graph, such that the total weight of the edges in E' is minimal and the graph $G'=(V,E-E')$ includes no cycles. Justify shortly the correctness of your algorithm.

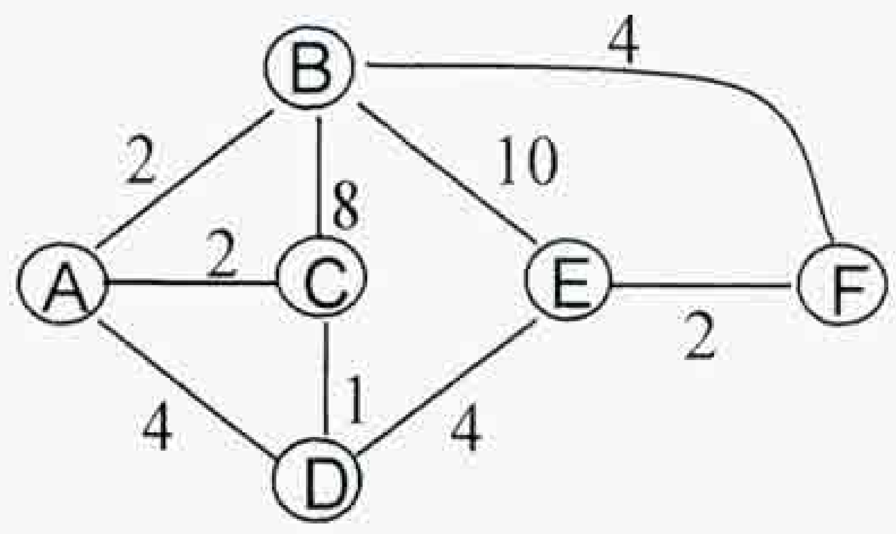
2. (10 pts.) Answer the following questions. No need to justify your answers.
The figure presents a flow network after three iterations of algorithm Ford & Fulkerson were performed.



a. What is the augmenting path performed in the first iteration?
b. Suggest an augmenting path in the next iteration
c. How many flow units can be routed in this path?
d. What is the value of the maximum-flow in the network?
e. Specify all sets of vertices that form a minimum cut in the network

Question 3: Computer Networks

Consider the network shown below:



Fill in the table below with the distance table that would be computed by the distance vector algorithm in node B, once the distance vector algorithm has finished executing. Assume poisoned reverse **is** used.

Table for B	destinations				
NEIGHBORS	A	C	D	E	F
A					
C					
E					
F					

Question 4: Computability and Complexity

a. Formally define the language A_{TM} and prove it is undecidable. (7 points)

b. Prove or refute: if $A \leq_p B$ and B is regular then A is regular. (6 points)

c. Let $L_1 = \{\langle M_1, M_2 \rangle \mid L(M_1) \cap L(M_2) \neq \emptyset\}$. Does $L_1 \in RE$? Prove your answer. (6 points)

d. Show that proving that 2SAT is not NP-complete would prove that 3SAT is not in P. (6 points)

Question 5: Operating System

The philosophers from the dining philosophers' problem have moved to live in a commune in which dinner is being held around a round table. At the center of the table there are n chopsticks (n is even), when n is the number of the philosophers. In the commune, philosophers take some two chopsticks from the center (that is, it does not matter which stick), and start eating. When they are done, they return the chopsticks back to the center of the table, think for a while, then repeating the process infinitely.

We are looking for an algorithm for the operation of the philosophers, such that a deadlock cannot happen in it, and not starvation. Also, it should allow performing as many operations in parallel as possible.

For each one of the solutions below, answer the following question.
(Assume that the semaphores used are fair, that is, they use a FIFO queue)

A. First solution:

```
// Init:
chopsticks = new Semaphore(n)
gate = new Semaphore(n)

gate.Down()
// Take chopsticks
chopsticks.Down()
chopsticks.Down()

// Eat

// Return chopsticks
chopsticks.Up()
chopsticks.Up()
gate.UP()
```

I. Does the solution answer the requirements? Explain and justify.

This question continues in the next page

B. Second solution:

```
// Init:
chopsticks = new Semaphore(n)
gate = new Semaphore(n/2)

// Take chopsticks
gate.Down()

chopsticks.Down()
chopsticks.Down()

// Eat

// Return chopsticks
chopsticks.Up()
chopsticks.Up()

gate.Up()
```

I. Does the solution answer the requirements? Explain and justify.

Question 6: Programming Languages

A. [16 points] In this part of the question you will explain the reasoning behind some of Java's syntax rules by giving examples of the negative consequences when the rules are violated. Each of the following Java code fragments contain a single syntax error. For each of them:

1. Explain where the syntax error occurs, and what general rule is violated (one line).
2. Write a fragment of Java code that includes the given fragment, has no additional syntax errors, and *demonstrates the problem that the syntax error is meant to avoid.* (i.e., if the compiler "ignored" the syntax error and you could run the code, it would cause a runtime error *due to this problem*).
3. Explain why the violation of the rule would cause a runtime error (in one or two lines).

Note: we are not looking for "trivial" syntax errors such as missing semicolons or `import` statements (you can assume that all the code surrounding the fragment is correct)
For the following fragments, assume the **A**, **B** and **C** are three classes such that **`B extends A`** and **`C extends B`**.

a. First fragment:

```
class X {
   public B getValue() {  return null;  }
}

class Y extends X {
   public A getValue() {  return null; }
}
```

b. Second fragment:

```
class Z {
   public List<A> getAList(List<B> x) { return x; }
}
```

B. [9 points] Write a short code fragment that behaves differently when it is evaluated in applicative order and when it is evaluated in normal order (you can write in pseudocode or in scheme). Give a one or two line explanation of what happens in each case. What is the evaluation order in Java?

GOOD LUCK