State Final Examination (Computer Science Sample Questions)

Fall 2021

1 Chomsky Hierarchy (3 points)

1. Give the definition of a context-free grammar and the language generated by a context-free grammar.
2. Classify the following language within the Chomsky hierarchy, that is, determine the smallest class (highest type) of the Chomsky hierarchy that the language belongs to and prove that it belongs to that class and does not belong to a smaller class.
$$L = \{a^i b^j \mid 0 \leq i \leq j \leq 2i\}$$

2 Minimum Spanning Trees (3 points)

1. Define the problem of finding a Minimum Spanning Tree (MST), and describe an algorithm solving MST; try to describe an algorithm with the best possible computational complexity. It is not necessary to describe in detail the data structures you use, but do describe the complexity of the operations of the respective data structures.
2. Prove the correctness of the algorithm you have described.
3. Describe an algorithm for the following problem with the best possible computational complexity. Given is a graph $G = (V, E)$, non-negative edge weights $(w_e)_{e \in E}$, a minimum spanning tree T, and a weight decrease of some edge $f \in E$, which is a new weight w'_f smaller than w_f. The task is to find a minimum spanning tree T' with respect to the updated weights, that is, an edge $e \neq f$ still has weight w_e but the weight of f is now w'_f instead of w_f.

   (In other words, the weight of one edge decreases arbitrarily and the task is to find a new minimum spanning tree, assuming a minimum spanning tree with respect to the old weights is known.)

3 SQL Querying (3 points)

Consider tables PERSON and CAR with schemas PERSON(ID, Name, Age) and CAR(ID, Color, OwnerID). The PERSON table contains the records ([1, Tom, 25], [2, Jane, 30], [3, Adam, 29], [4, Max, 36]). The CAR table contains the records ([1, Red, 4], [2, Blue, 1], [3, Red, 2], [4, White, 4]).

Write the result of the following SQL queries, or mark the location of an error if a SQL query is not correct.

1. Select Name From PERSON Inner Join CAR On (ID = ID)
2. Select Max(Age) From PERSON Where Age < 30
3. Select Name From PERSON Where Age = Max(Age)
4. Select Name From PERSON Where ID Not In (Select OwnerID From CAR)
5. Select Color, Count(ID) From CAR Where ID = 1 Group By Color Having Count(ID) > 1
6. Select * From CAR Where OwnerID >= ALL (Select ID From PERSON)

4 Object Design of a Card Game (3 points)

Suppose we want to implement a multiplayer turn-based game as a computer version of a card game from a world ruled by magic. The basic card of this game is a **creature** type card – each creature has: **a) power** = a non-negative integer, **b) a**

list of abilities, which can be empty or contains just a small amount of items. The power and contents of the ability list for each card present in the game can be changed with different spells during the game.

Ability is a boolean property characterized only by whether a creature has it or not. A creature may have a specific ability in the list more than once – but for a potential evaluation of the effects of the ability, it is equivalent to having it in the list just once.

Choose the C#, C++, or Java programming language, note your choice, and write all parts of the solution in the chosen language.

Part A

Write an implementation of the `Creature` class that represents a creature card in the game. Write only a public contract declaration of such a class and do not address the implementation details of individual methods and private fields. At the same time, design and describe any other (non-trivial) types needed to design the `Creature` class contract. Your solution must support at least: **a) adding** an ability to the creature, **b) removing** one instance of an ability, i.e. if the creature has the ability more than once in its list, the number of occurrences will be reduced by 1, **c) checking** if the creature has an ability or not.

The ability support must be designed in an extensible way so that we can add new abilities to the game without modifying the original game code (for example, in a library with a game extension). At the same time, the ability system should be designed as strongly typed as possible, so that a possible misspelling of an ability name in the algorithms evaluating the effects of abilities leads to a compile time, rather than runtime, error.

Use the proposed infrastructure to write an implementation/declaration of a "*defender*" ability, and a "*lifelink*" ability.

Next, write an example of creating a `Creature` instance representing a "*Gwynevere*" creature with the initial power of 2 and the initial abilities "*defender*" and "*lifelink*".

Part B

Another type of card in the implemented game is the **enchantment** card. The enchantment card can be magically attached to a suitable creature card during play – the enchantment card is attached to the target creature card until it is removed by another spell. Each type of enchantment card includes an "algorithm" that checks whether the enchantment card is applicable to a particular creature card.

Extend the implementation of the `Creature` class with the enchantment support (and add any other types needed) and write an example of an "*Angelic Gift*" enchantment card implementation that can only be applied to creatures with a power of less than 10, and only if the creature also has the "*defender*" and "*lifelink*" abilities.

Part C

If an enchantment card is attached to a creature card, then the attachment event has some effect on the target creature given by the type of the enchantment – this effect can in principle arbitrarily modify the properties of the given creature. Complete your implementation so that when an enchantment is attached to a creature, **the effect is applied**, and when an enchantment is disconnected, **the effect is canceled**.

Extend the implementation of the "*Angelic Gift*" enchantment card to add 3 power points and the "*flying*" ability to the target creature (assume that you already have the "*flying*" ability defined in a similar way to the "*defender*" and "*lifelink*" abilities). After disconnecting the "*Angelic Gift*" from the creature, this creature loses 3 power points again, as well as the "*flying*" ability (unless it came from another source, or as an initial ability).

Then write **a)** an example of a code that attaches the "*Angelic Gift*" card to the "*Gwynevere*" creature card, and **b)** an example of a code that removes the "*Angelic Gift*" card from the "*Gwynevere*" creature card.

5 Computer Architecture (3 points)

A storage controller provides a register-based interface for accessing disk drives. The interface consists of several 8-bit registers that are mapped to a block of memory starting at address `0xE00007F8`. The register layout (including the offset from the base address) and the meaning of individual registers is illustrated in the following picture:

The meaning of relevant bits of the *Status* register is shown in the following picture:

In the following, assume that all communication and data transfer is performed in **PIO** (*programmed I/O*) mode, i.e., **without** interrupts or direct memory access (DMA). The following fuctions are available to safely access the registers of the controller:

	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
+0 B	Data								Error							
+2 B	Count								LBA[7:0]							
+4 B	LBA[15:8]								LBA[23:16]							
+6 B	Drive				LBA[27:24]				Status							

Register	Meaning
Data	Register for transferring data to/from disk.
Error	Error code, if the device indicates an error.
Count	Number of sectors in a read/write request.
LBA	The number of the first sector in a read/write request (28-bit logical block address). The [m:n] range indicates LBA bits that are accessible through the register.
Drive	The number of the device the controller should communicate with.
Status	Indicates device and request status.

	7	6	5	4	3	2	1	0
Status	BUSY				DRQ			ERROR

Bit	Meaning
BUSY	Indicates that the device is working (and cannot accept commands).
DRQ	Indicates readiness to transmit data of a single sector (Data Request Ready).
ERROR	Indicates that an error occurred while processing a request.

```
uint8_t io_read_byte(uintptr_t address);
void io_write_byte(uintptr_t address, uint8_t value);
```

The `address` and `value` parameters correspond to a register address and the value that should be written to the register. The `uintptr_t` type represents unsigned integers with a range sufficient for storing an address.

Part A

Assume that the controller only allows connecting disk drives with **512**-byte sectors. Answer the following questions given the information about the controller interface:

1. What is the maximum size of a read/write request?
2. What is the maximum disk drive capacity that we can work with?

Justify your answers and state the values in bytes (with a suitable binary unit prefix).

Part B

Write a function (in pseudocode) that uses the controller interface to set up the drive for a single request. The function has the following signature:

```
void setup_request(uint8_t drive, uint32_t lba, uint8_t count);
```

The `drive`, `lba`, and `count` parameters correspond to the device number, the number of the first sector of the request, and the number of sectors in the request, respectively. Assume that the device is ready to communicate, and that the activation of the request is done elsewhere (i.e., the `setup_request` function does not need to deal with it).

Part C

Write a function that transfers data of recently finished read request from disk to memory. The disk transfers data sector by sector and indicates its readiness to transfer data of a single sector by setting the `DRQ` bit to `1` and the `BUSY` bit to `0`. If no error occurred while reading the sector, the `ERROR` will be set to `0`, otherwise it will be set to `1` and the request will be aborted.

Because the data transfer is performed in PIO mode, the data needs to be read byte-by-byte from the *Data* register. After the last byte of a sector has been transferred, the `DRQ` bit will be set to `0` and the drive may become busy (indicating activity using the `BUSY` bit) before making the data of the next sector available. It is therefore necessary to wait until the drive is ready to transfer the next sector and ensure that no error occurred. The function has the following signature:

```
unsigned int read_sectors(unsigned int count, uint8_t * buffer);
```

The `count` parameter represents the **number of sectors** that should be read, and the `buffer` parameter is a pointer to an array of bytes into which the data of all sectors should be stored. The return value indicates the number of sectors read and can be lower than the number of sectors requested if an error occurred. It is sufficient to check for an error before starting the transfer of each sector.

6 Transport Protocols and Their Properties (3 points)

Explain the following terms and briefly enumerate what we want to achieve using them/what problems we have to figure out to guarantee them.

1. Stream transmission
2. Connection-oriented transmission
3. Reliable transmission

Propose and thoroughly describe how would you implement these features in a hypothetical (yet meaningfully and effectively operating) transport layer L4 protocol. You can get an inspiration from the existing protocols in the TCP/IP architecture, or you can design your own.

7 Optimization Methods (3 points)

The carving workshop Pech and Son plans to focus on two products in the first week of April: a giant and a wolf.

Producing a giant requires a cuboid of linden wood measuring 250 x 120 x 340 mm, which costs 800 CZK; producing a wolf requires a cuboid of alder wood measuring 250 x 100 x 270 mm, which costs 600 CZK. Carving a giant takes four hours (no matter what carver from the workshop does the job, they are all equally skilled); carving a wolf takes five hours. Giants and wolves must be surface-treated before sale, which takes two hours for a giant and one hour for a wolf. Four carvers work full-time in the workshop, each of them 35 hours a week. The carving workshop has at its disposal 20 000 CZK for purchasing wood, and it has a week to do the carving. Surface treatment is outsourced to external collaborators, who have 48 hours available for doing the work. One giant has a selling price of 2050 CZK, and one wolf sells for 2100 CZK.

The aim is to plan the production so that the total selling price of the giants and wolves combined is as high as possible, subject to the above restrictions (i.e., price of purchased material, time constraints of carvers and external collaborators).

1. Formulate the problem as an integer linear program.
2. Solve the linear relaxation of your formulation from part 1 (a graphical method is recommended).
3. Find the integer optimum.

8 Aproximační algoritmy (specialization question – 3 body)

1. Zaveďte pojem “aproximační faktor algoritmu”.
2. Na problému rozvrhování úloh na identických strojích ilustrujte techniku lokálního prohledávání (tj. popište problém a algoritmus, který problém předepsanou technikou řeší, uveďte aproximační poměr algoritmu).
3. Analyzujte výše popsaný algoritmus (tj. dokažte správnost, odhad aproximačního poměru a časové složitosti).

9 Codes (specialization question – 3 points)

Define a perfect code and give an example.

Is there a perfect binary code of length 9 and distance 3? Give an example or show why such a code does not exist.

10 Chromatic Index (specialization question – 3 points)

Define the chromatic index (edge coloring number) of a graph.

Let G be a cubic (3-regular) connected graph. Which of the following claims hold and why?

1. If G has a perfect matching, then it is 3-edge-colorable.
2. If G is 3-edge-colorable, then it has a perfect matching.
3. If G has a Hamiltonian cycle, then it is 3-edge-colorable.
4. If G is 3-edge-colorable, then it has a Hamiltonian cycle.

11 Well-Ordering (specialization question – 3 points)

1. (a) Write a definition of well-ordering. It is not necessary to define the notion of *ordering* itself.

   (b) Write a definition of ordinal number. If you use the notion of a *transitive set*, define it as well.

2. Which of the following sets are ordinal numbers? Briefly explain each answer.

   (a) 2021

   (b) $\bigcup 2021$

   (c) $20 \cap 21$

   (d) ω

   (e) $\{\omega\}$

   (f) $\omega \cup \{\omega\}$

   (g) $\omega \cap \{\omega\}$

   (h) $\omega \cup \{\omega\} \cup \{\{\omega\}\}$

12 Morphological, Syntactic and Semantic Analysis of Natural Languages (specialization question – 3 points)

1. Explain the following terms: *stemming*, *lemmatization*, *tagging* and *morphological analysis*. Give some examples.
2. Explain the term *Universal Dependencies*.

13 Basic Formalisms for Description of a Natural Language (specialization question – 3 points)

1. How does the Prague Dependency Treebank (PDT) relate to the Functional Generative Description?
2. Draw a syntactic and tectogrammatical tree in the PDT format for a (suitable) sentence of your choice. Describe their differences.

14 Basic Information Theory (specialization question – 3 points)

Let X and Y be two random variables, H stands for Entropy. Decide whether the relations below are **true** or **false**. If **true**, provide a proof. If **false**, give a counterexample or prove that the opposite is true. Hint: use the relationship between Mutual information and Entropy.

1. $H(X) \geq H(X|Y)$
2. $H(X) + H(Y) \leq H(X,Y)$