State Final Examination (Sample Questions)

2024-06-25

1 Divide and conquer algorithms (shared topics)

1) Consider algorithm A which performs $T(n)$ elementary operations on data of size n. Algorithm A is recursive and works as follows (a, c, x, y are natural numbers, $a > 0$, $c > 1$):

a) Performs $\Theta(n^x)$ elementary operations to select a subsets of size n/c from the input data.

b) Recursively runs itself on each of the selected subsets of data (if the size is at least c, for smaller data solves the problem in constant time).

c) Performs $\Theta(n^y)$ elementary operations to unify all solutions obtained in b) into a solution of the original problem on data of size n.

Determine (without a proof) an asymptotically tight estimate of $T(n)$ depending on parameters a, c, x, y.

2) Let X and Y be two arrays of length n, each containing a *sorted* sequence of n natural numbers. Design and describe an algorithm with an $O(\log n)$ time complexity, which finds a median (one of the two medians) of all $2n$ numbers contained in arrays X and Y. Using the method from 1) prove that the time complexity of your algorithm is indeed $O(\log n)$.

Solution sketch 1) The given problem is solvable using the Master Theorem. The combined overhead in A is $\Theta(n^x + n^y) = \Theta(n^{\max\{x,y\}})$ and is performed in each recursive call. This gives us a recursive equation $T(n) = a \cdot T(n/c) + \Theta(n^{\max\{x,y\}})$ for time complexity of A which is solvable using the Master Theorem. Let $d = \max\{x, y\}$. The solution has 3 cases:

1. $T(n) = \Theta(n^d)$ if $a/c^d < 1$;
2. $T(n) = \Theta(n^d \log n)$ if $a/c^d = 1$;
3. $T(n) = \Theta(n^{\log_c a})$ if $a/c^d > 1$.

2) We compare medians of both arrays X and Y in each step. The medians can be e.g. elements $x = X[\lfloor (n-1)/2 \rfloor]$ and $y = Y[\lfloor (n-1)/2 \rfloor]$ using 0-based indexing.

If $x < y$, we can discard a left half of X (elements to the left of x) and a right half of Y (elements to the right of y) and call a recursion for the remaining elements. It is very important that the number of elements discarded from X (which are all guaranteed to be smaller then the median we are looking for) is exactly the same as the number of elements discarded from Y (which are all guaranteed to be greater then the median we are looking for). This property ensures that the median of the elements that enter the recursive call is the same as the median of the original set of elements.

If $x > y$, then we discard symmetrically a left half of Y and a right half of X and we call a recursion.

If $x = y$, then x is a median (one of them).

Retrieving x and y, their comparison, and index shifts (of current boundaries in X and Y) takes constant time $\Theta(1) = \Theta(n^0)$. We compute a median in constant time for $n < 3$.

The time complexity of algorithm is described by an equation $T(n) = 1 \cdot T(n/2) + \Theta(1)$, i.e. $a = 1$, $c = 2$, $d = 0$. Using a Master Theorem we get case 2, because $a/c^d = 1/2^0 = 1$, and the solution $\Theta(n^0 \log n) = \Theta(\log n)$.

2 Semaphore (shared topics)

The C# `System.Threading` library provides the class `Semaphore`, with functions corresponding to the classical semaphore P (`WaitOne`) and V (`Release`) operations. When created with `new Semaphore(0, 1)`, it corresponds to a binary semaphore initialized to zero. If more than one thread is waiting for the same semaphore, the order in which the threads are released when the semaphore is signalled is not specified.

The program on the right shall produce the output `OneTwoThreeFourFive`; however, the odd parts of the output shall be written by function `f1` and the even parts by function `f2`, these run in different threads.

The synchronization, implemented using a semaphore, almost works, but contains race conditions.

1. Describe a scenario that produces an output different from `OneTwoThreeFourFive` or ends in a deadlock. Use a sequence of line numbers to denote a scenario, assuming that a number denotes *finishing* the statement at that line. Only the code of `f1` and `f2` is relevant.
2. Would the race condition still exist if the C# library guaranteed FIFO order of releasing threads from `WaitOne`?
3. Write a race-free solution of the problem that reliably produces the required output `OneTwoThreeFourFive`. Use two semaphores and only their `Release()` and `WaitOne()` functions. Your solution should rely on the two semaphores only, no other variables or objects should be shared between the threads. The `Console.Write` calls must naturally remain where they are.

(The question uses the C# language merely as a representative of general purpose programming languages, neither the question nor the solution depends on any C# language features.)

```
1  class Program
2  {
3      private static Semaphore sem;
4
5      private static void f1()
6      {
7          Console.Write("One");
8          sem.Release();
9          sem.WaitOne();
10         Console.Write("Three");
11         sem.Release();
12         sem.WaitOne();
13         Console.Write("Five");
14     }
15
16     private static void f2()
17     {
18         sem.WaitOne();
19         Console.Write("Two");
20         sem.Release();
21         sem.WaitOne();
22         Console.Write("Four");
23         sem.Release();
24     }
25
26     static void Main(string[] args)
27     {
28         sem = new Semaphore(0, 1);
29         Thread t1 = new Thread(f1);
30         Thread t2 = new Thread(f2);
31         t1.Start();
32         t2.Start();
33         t1.Join();
34         t2.Join();
35     }
36 }
```

Solution sketch

1. All scenarios, in lexicographical order:
   (a) 7-8-9-10-11-12-13-deadlock(18+34): `OneThreeFive`
   (b) 7-8-9-10-11-18-19-20-12-13-deadlock(21+34): `OneThreeTwoFive`
   (c) 7-8-9-10-11-18-19-20-21-22-23-12-13: `OneThreeTwoFourFive`
   (d) 7-8-18-19-20-9-10-11-12-13-deadlock(21+34): `OneTwoThreeFive`
   (e) *7-8-18-19-20-9-10-11-21-22-23-12-13:* ***`OneTwoThreeFourFive`*** *- The expected program behavior*
   (f) 7-8-18-19-20-21-22-23-9-10-11-12-13: `OneTwoFourThreeFive`

   Any of the above scenarios except (e) is a sufficient answer.
2. Yes. All the scenarios above work also for FIFO-waits.
3. See code below.

```
private static void f1()
{
```

```
    Console.Write("One");
    semA.Release();
    semB.WaitOne();
    Console.Write("Three");
    semA.Release();
    semB.WaitOne();
    Console.Write("Five");
}

private static void f2()
{
    semA.WaitOne();
    Console.Write("Two");
    semB.Release();
    semA.WaitOne();
    Console.Write("Four");
    semB.Release();
}
```

3 Vector space bases (shared topics)

1. Formulate the Steinitz's theorem about the exchange of suitable vectors between linearly independent and generating sets (not necessarily bases).
2. Consider a real matrix $\boldsymbol{A}$ such that it is similar to a diagonal matrix $\boldsymbol{D}$ via the product $\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{R}$, where

$$\boldsymbol{R} = \begin{pmatrix} -1 & -1 & -8 & -2 & -8 \\ 0 & 1 & -3 & 0 & 2 \\ -1 & -2 & 6 & -2 & 3 \\ -1 & -3 & 7 & -1 & -4 \\ 1 & 2 & 0 & 2 & 4 \end{pmatrix},$$

   where it is known that the elements on the diagonal $\boldsymbol{D}$ are the numbers $17, 17, 23, 17, 23$ in this order.

   Decide whether any of the columns of $\boldsymbol{R}$ can be replaced by the following vector $\boldsymbol{u_i}$, so that the matrix $\boldsymbol{A}$ can still be diagonalized through the resulting matrix $\boldsymbol{R}'$.

   If such an exchange is possible, determine all columns of the matrix $\boldsymbol{R}$ that can be exchanged.

   (a) Solve for $\boldsymbol{u_1} = (4, 2, 2, -3, -2)^{\mathrm{T}}$.

   (b) Solve for $\boldsymbol{u_2} = (7, -6, -2, 12, -6)^{\mathrm{T}}$.

   Justify your answers.

Solution sketch

1. Let B be a finite linearly independent set in vector space V and let C generates V. Then there exists D such that: D generates V, $B \subseteq D$, $|D| = |C|$ and $D \setminus B \subseteq C$.
2. Given the basis B of the columns $\boldsymbol{R}$ are (computed by solving the systems):
   (a) $[\boldsymbol{u}_1]_B = (0, 2, 0, -3, 0)^{\mathrm{T}}$ so $\boldsymbol{u}_1$ can be replaced by the 2nd and 4th columns, they correspond to the same eigenvalue 17, so the eigenvector subspace is unchanged.
   (b) $[\boldsymbol{u}_2]_B = (0, 1, 1, 0, -2)^{\mathrm{T}}$ or $\boldsymbol{u}_2$ does not lie in any subspace of the eigenvectors, so it is not an eigenvector and therefore cannot be exchanged.

4 Model of a theory (shared topics)

1. Let T be a theory of language (signature) $L = \langle \mathcal{R}, \mathcal{F} \rangle$ in predicate logic (where $\mathcal{R}, \mathcal{F}$ are sets of relation and function symbols with given arities). Give definitions of a *structure of language* L and a *model of theory* T.
2. Write the following statements by formulas $\varphi_1, \varphi_2, \varphi_3$ in language $L = \langle E, L, P \rangle$ of predicate logic (with unary predicates for *'pass the exam', 'be lucky', 'be prepared'*).
   (a) *Not everyone who passes the exam is lucky, but who is lucky passes the exam.*
   (b) *Luck favors the prepared. (Who is prepared is lucky.)*
   (c) *Some student was prepared but did not pass the exam.*
3. For the theory $T_1 = \{\varphi_1, \varphi_2\}$ and also for the theory $T_2 = \{\varphi_1, \varphi_2, \varphi_3\}$, either find a model of the theory or prove formally (using tableu method, resolution, or Hilbert's calculus) that the theory does not have a model.

Solution sketch

1. A structure of language L is a triple $\mathcal{A} = \langle A, \mathcal{R}^A, \mathcal{F}^A \rangle$ where A is a nonempty set (domain) and $\mathcal{R}^A$, $\mathcal{F}^A$ are collections of realizations of relation and function symbols of language L. A realization of an n-ary relation symbol R is some n-ary relation $R^A \subseteq A^n$, a realization of an n-ary function symbol f is some n-ary function $f^A \colon A^n \to A$, where for a realization of a constant symbol c (as a nullary function symbol) we take directly some element $c^A \in A$.

   Note: We can also accept an answer in the sense that a structure is some k-tuple with nonempty domain and realizations of corresponding relation and function symbols, with explanation what these realizations are.

   A model of a theory T is a structure of language L where all axioms of T are valid.
2. For example
   (a) $\varphi_1 : \neg(\forall x)(E(x) \to L(x)) \wedge (\forall y)(L(y) \to E(y))$
   (b) $\varphi_2 : (\forall z)(P(z) \to L(z))$
   (c) $\varphi_3 : (\exists s)(P(s) \wedge \neg E(s))$
3. Theory T_1 has for example a singleton model $\mathcal{A} = \langle \{0\}, E^A = \{0\}, L^A = P^A = \emptyset \rangle$. Theory T_2 does not have model, which can be proved by contradiction from T_2 using the tableu method or Hilbert's calculus, or by refutation of the skolemised theory T_2 transformed into CNF using resolution.

5 Data model (specialization DW)

The following logical relational data model is used in a hospital information system (keys are underlined and foreign keys are italicized):

- Diagnosis(**Code**,Name)
- Doctor(**SSN**, Name, Surname, City, Street, Number, Zipcode, Specialization, YearsOfPractice)
- Patient(**SSN**, Name, Surname, City, Street, Number, Zipcode, *GeneralPractitioner*), GeneralPractitioner $\subseteq$ Doctor[SSN]
- TreatedFor(***SSN***, *Code*), SSN $\subseteq$ Patient[SSN], Code $\subseteq$ Diagnosis[Code]

1. Represent the logical relational data model above using a diagram in your chosen conceptual language (ER, UML).
2. If needed, extend the conceptual model and the corresponding logical relational data model so that:
   - each patient can be treated for any number of diagnoses,
   - each patient has exactly one general practitioner and can have any number of other doctors,
   - each patient can undergo any number of examinations for a diagnosis and vice versa,
   - a patient who is also a doctor should not have her/his personal data recorded redundantly.

Solution sketch

1. Three entities (classes): Diagnosis, Doctor, Patient. One mandatory 1:N relationship between the general practitioner and the patients. One optional 1:N relationship between diagnosis and patients.
2. – The TreatedFor association table already exists, but supports only the 1:N cardinality. Just change the key definition to TreatedFor(***SSN***, ***Code***), SSN ⊆ Patient[SSN], Code ⊆ Diagnosis[Code].
   – Patients already have just one general practitioner. Add an M:N relationship and a corresponding association table for the M:N relationship between doctors and patients, Treats(***PatientSSN***, ***DoctorSSN***), PatientSSN ⊆ Patient[SSN], DoctorSSN ⊆ Doctor[SSN].
   – M:N association Examination between patient and diagnosis, and the corresponding association relation Examination(***PatientSSN***, ***Code***), PatientSSN ⊆ Patient[SSN], Code ⊆ Diagnosis[Code].
   – In the simplest solution, the Doctor class will be a subclass of the Patient class, or both will be a subclasses of a new Person class. In the first case, the Doctor relation will lose the attributes Name ... ZIPCode and SSN will be a foreign key to Patient[SSN]. In the second case, the new class Person will contain the attributes SSN ... ZIPCode, and both Patient and Doctor will lose the attributes Name ... ZIPCode, SSN will be a foreign key to Person[SSN].

6 Transactions (specialization DW)

For the following schedule, where R denotes a read operation and W denotes a write operation:

	T_1	T_2	T_3
1	W($Patient_1$)		
2		???	
3		W($Patient_1$)	
4		COMMIT	
5			R($Patient_1$)
6			W($Patient_1$)
7			COMMIT
8	R($Patient_2$)		
9	COMMIT		

1. What database read/write operation in transaction T_2 at time 2 in place of ??? will make the schedule not conflict-serializable? Justify your decision.
2. What database read/write operation in transaction T_2 at time 2 in place of ??? will make the schedule unrecoverable. Justify your decision.

Solution sketch

– W($Patient_2$). A cycle occurs in the precedent graph $T_1 \rightarrow T_2$ and $T_2 \rightarrow T_1$.

– R($Patient_1$). A read of the uncommitted value written by transaction T_1, but COMMIT precedes the end of transaction T_1.

7 API and Scripting (specialization DW)

1. Design and describe the REST API for the hospital system from the earlier question. The API must support the following functionality:
   – to retrieve patients whose name matches the query pattern,
   – retrieving information about a specific patient,
   – create, update and remove patient,

- get a list of patient diagnoses,
- add, remove patient diagnosis,
- get list of patient's doctors,
- get a list of patients who have at least one of the specified diagnoses (the list of diagnoses can be very long).

2. Let's have a web application that provides access to the IS through the proposed API. Write a JavaScript snippet that, when the query patter (first endpoint) is entered and a button is clicked, invokes the corresponding HTTP API request, retrieves all patients matching the query and displays them in a list. When the user clicks on a particular patient from the list retrieved by the first query, the list of its doctors and diagnoses is updated by retrieving the data from the API and then displaying it.

   Assume that the API returns data in JSON format. Briefly describe the format by a short example for each of the endpoints used. Assume the existence of `displayDoctors` and `displayDiagnoses` functions to provide DOM tree editing capabilities. Similarly, the `clearDoctors` and `clearDignoses` functions are available. Further, assume the existence of all necessary HTML elements.

   Pay attention to the proper use of asynchronous request processing. Ensure that even in the event of adverse concurrency of asynchronous calls, after loading the list of patients, the lists of doctors and diagnoses remain in a consistent state, i.e., the lists are displayed for the selected patient and are not displayed when no patient is selected. It is not necessary to handle fetch operation error states and authentication.

Solution sketch

1. The corresponding endpoints should look like the following, for example
   (a) GET api/v1/patients?query={pattern}, or possibly GET api/v1/patients/{pattern}
   (b) GET/DELETE/POST api/v1/patient/{rc}
   (c) GET api/v1/patient/{rc}/diagnoses
   (d) GET/POST/DELETE api/v1/patient/{rc}/diagnoses
   (e) GET api/v1/patient/{rc}/doctors
   (f) POST api/v1/diagnoses

   Since the list of diagnoses in the last API endpoint can be long, the payload needs to be sent in the body of the query and thus a GET request cannot be used for this endpoint.

2. For example:

```
let currentQuery = '';

document.getElementById('searchButton').addEventListener('click', async () => {
  const query = document.getElementById('searchInput').value();
  currentQuery = query;

  const patientList = document.getElementById('patient-list');

  patientList.innerHTML = '';
  clearDoctors();
  clearDiagnoses();

  const patients = await (
    await fetch(`/api/v1/patients?query=${encodeURIComponent(query)}`)
  ).json();

  patients.forEach(patient => {
    const item = document.createElement('li');
    item.textContent = patient.name;
```

```
    item.dataset.rc = patient.rc;
    item.addEventListener('click', () => fetchPatientDetails(patient.rc, query));
    patientList.appendChild(item);
  });
}

async function fetchPatientDetails(rc, query) {
  const doctors = await (await fetch(`api/v1/patient/${rc}/doctors`)).json();
  const diagnoses = await (await fetch(`api/v1/patient/${rc}/diagnoses`)).json;
  if (query != currentQuery){
    displayDoctors(doctors);
    displayDiagnoses(diagnoses);
  }
}
```

3. For example, the API to retrieve the patient list will have the form

   `[{ "rc": <person number>, "name": <name>, "surname": <surname>, ... }]`

4. The solution should demonstrate iteration, use of async/await and working with data attributes (list items).
5. The solution should guarantee that when the searchButton listener is called and results from the Doctors and Diagnoses API are pending, these do not overwrite the already cleaned elements.

8 Information retrieval (specialization DW)

Consider a situation where the system from the earlier questions needs to be enriched to allow a diagnosis to be made based on the symptoms entered. Let us have a collection of documents where each document describes a specific disease.

1. What would a boolean model look like for such a task? What terms would a specialized dictionary for this task contain? How are queries and documents represented in the Boolean model, and how can querying be implemented efficiently in this model?
2. Let's say we want to modify the system so that the query is not a list of symptoms, but directly a report of a medical examination. What modification of the Boolean model is appropriate to use in this case? How will the representation of the query and the document change? How will the similarity between the query and the document be evaluated then and why?

Solution sketch

1. Boolean model
2. (a) Terms = symptoms, documents = descriptions of diseases
   (b) Specialised symptoms dictionary, i.e., no need to retrieve the dictionary as in classical DIS by parsing documents
   (c) System represented by a binary matrix, queries implemented as bitwise operations on a vector of terms (symptoms)
   (d) Efficient representation by inverted index and merge sort-like processing
3. Extensions – e.g. vector model
4. (a) The query will be represented in the same way as the document
   (b) Similarity is typically evaluated using the cosine distance in tf-idf space between the query and document vectors

9 Error-correcting codes (specialization OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

In this question we only work with binary error-correcting codes, i.e., ones over the alphabet $\{0, 1\}$.

1. Define the following terms: Hamming distance, error-correcting code, the length, size, and distance of an error-correcting code.
2. Let C be an error-correcting code of size $k \geq 2$ and length n. Let x be a word over the alphabet $\{0,1\}$ of length m. How would the code C be used to encode the word x so that the resulting word is as short as possible? Determine the length of the resulting word, depending on the parameters k, n, and m.
3. Is there an error-correcting code of length 5, size 6, and distance 3? If so, construct one; if not, prove that it does not exist.

Solution sketch

1. The *Hamming distance* $d(x,y)$ of two words x and y of the same length is the number of positions where x and y differ. An *error-correcting code* is an arbitrary set of words of the same length, i.e., in the binary case, an arbitrary subset $C \subseteq \{0,1\}^n$. Here n is a positive integer called the *length* of the code C. The *size* of the code C is defined as the size $|C|$ of the set C, i.e. as the number of its words. The *Distance* of the code C is the minimum of the Hamming distances between its words, i.e.,
$$\min_{x,y \in C, x \neq y} d(x,y).$$
2. One possibility is to interpret x as a number in the range from 0 to $2^m - 1$ written in binary. We can express this number in base k as a sequence of digits $q_0 q_1 \dots q_{t-1}$, where $t = \lceil \log_k 2^m \rceil$. Next, we fix a bijection $c : \{0, \dots, k-1\} \to C$. The resulting word will then be the concatenation of the words $c(q_0)$, $c(q_1)$, …, $c(q_{t-1})$. The length of this word is
$$n \cdot \lceil \log_k 2^m \rceil = n \cdot \left\lceil \frac{m}{\log_2 k} \right\rceil \approx \frac{n}{\log_2 k} \cdot m.$$
Another option is to divide x into blocks of $\lfloor \log 2k \rfloor$ bits and encode each block separately. However, when k is not a power of 2, this leads to a suboptimal length $\approx \frac{n}{\lfloor \log 2k \rfloor} \cdot m$ of the resulting word.
3. Such a code cannot exist, as it does not meet the condition of the Hamming bound. Alternatively, one can prove this directly: In a code C of distance 3, the balls of radius 1 (in Hamming distance) around each of its words are disjoint. If the code C has length 5, each of these balls has size 6, and thus no more than
$$\left\lfloor \frac{2^5}{6} \right\rfloor = 5$$
of them can fit into the space $\{0,1\}^5$ of words.

10 Subgraphs with prescribed degrees (specialization OI-G-PDM, OI-PADS-PDM)

1. Define the notion of perfect matching in a general graph, and formulate the necessary and sufficient condition for its existence (Tutte's theorem).
2. Let G be a graph and let $d : V(G) \to \mathbb{N}$ be an arbitrary function. Let H be the graph obtained from G by replacing each vertex v with a copy of the complete bipartite graph $K_{\deg v, \deg_G v - d(v)}$ with parts A_v and B_v, and by adding a perfect matching on vertices $\bigcup_{v \in V(G)} A_v$ such that for each edge $\{u,v\} \in E(G)$, there is exactly one edge in H between A_u and A_v. More precisely,
$$\begin{aligned} A_v &= \{(v,u) : \{v,u\} \in E(G)\} && \text{for each } v \in V(G) \\ B_v &= \{(v,i) : v \in V(G), i \in \{1, \dots, \deg_G(v) - d(v)\}\} && \text{for each } v \in V(G) \\ V(H) &= \bigcup_{v \in V(G)} (A_v \cup B_v) \\ E(H) &= \{\{(u,v),(v,u)\} : \{u,v\} \in E(G)\} \cup \bigcup_{v \in V(G)} \{\{x,y\} : x \in A_v, y \in B_v\} \end{aligned}$$
Show that the graph H has a perfect matching if and only if G contains a subgraph G' such that $V(G') = V(G)$ and each vertex v has degree exactly $d(v)$ in G'.

3. Design a polynomial-time algorithm that, for an input consisting of a graph G and a function $d : V(G) \to \mathbb{N}$, finds a subgraph $G' \subseteq G$ such that $V(G') = V(G)$ and every vertex v has degree exactly $d(v)$ in G', or decides that no such subgraph exists.

Solution sketch

1. *Perfect matching* in a graph G is a subset of M of its edges such that every vertex of G is incident with exactly one of the edges of M. There are other possible equivalent definitions, e.g., a 1-regular spanning subgraph of G.

   Tutte's theorem: A graph G has a perfect matching if and only if

   $$\operatorname{odd}(G - S) \leq |S|$$

   for every subset $S \subseteq V(G)$, where $\operatorname{odd}(F)$ is the number of components of the graph F that have an odd number of vertices.

2. If there is such a subgraph G', then $M = \{\{(u, v), (v, u)\} : \{u, v\} \in E(G')\}$ is a matching in H covering for each $v \in V(G)$ exactly $d(v)$ vertices of A_v. The remaining $\deg_G v - d(v)$ vertices from A_v can be arbitrarily matched with the vertices of B_v, thereby extending M to a perfect matching in H.

   Conversely, let M' be a perfect matching in H. For each $v \in V(G)$, this matching contains exactly $|B_v| = \deg_G v - d(v)$ edges between B_v and A_v, and therefore exactly $d(v)$ edges with one end in A_v and the other end outside of B_v. Hence the subgraph

   $$G' = (V(G), \{\{u, v\} : u, v \in V(G), \{\{(u, v\}, (v, u\} \in M'\})$$

   satisfies the prescribed conditions for the degrees of vertices.

3. Simply apply (e.g.) Edmonds' algorithm to find a perfect matching M' in the graph H, and if one exists, use the transformation described in the previous paragraph.

11 Strongly connected component (specialization OI-G-PADS, OI-O-PADS, OI-PADS-PDM)

1. Define *strongly connected components* of a directed graph. With what time complexity can you find them? (You need not describe details of the algorithm.)
2. Define the *component graph* (a.k.a. condensation) and state its properties.
3. Design an algorithm that is given a directed graph with some vertices marked green. The output should be a walk (not necessarily a path) containing as many green vertices as possible. The algorithm should run in polynomial, preferably linear time.

Solution sketch

1. See Průvodce labyrintem algoritmů, section 5.9.
2. Dtto.
3. When a walk visits a strongly connected component, it can collect all green vertices in it. We therefore construct the component graph and label each of its vertices with the number of green vertices in the corresponding component. In this graph, we want to find the walk with maximum sum. Since the component graph is acyclic, the walk will be a path and it can be found by induction on the topological order. The whole algorithm runs in linear time with respect to the number of vertices and edges of the graph.

12 Multivariate functions (specialization OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

1. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space and let $d_n(x, y)$ denote the Euclidean distance of points $x, y \in \mathbb{R}^n$. Given a function $f \colon \mathbb{R}^m \to \mathbb{R}^n$ and a point $x \in \mathbb{R}^m$, define what it means to say that f *is continuous in* x.

2. Consider the function $f(x, y)\colon \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f(x, y) = \exp(x^2) \cdot \ln(2 + y^2),$$

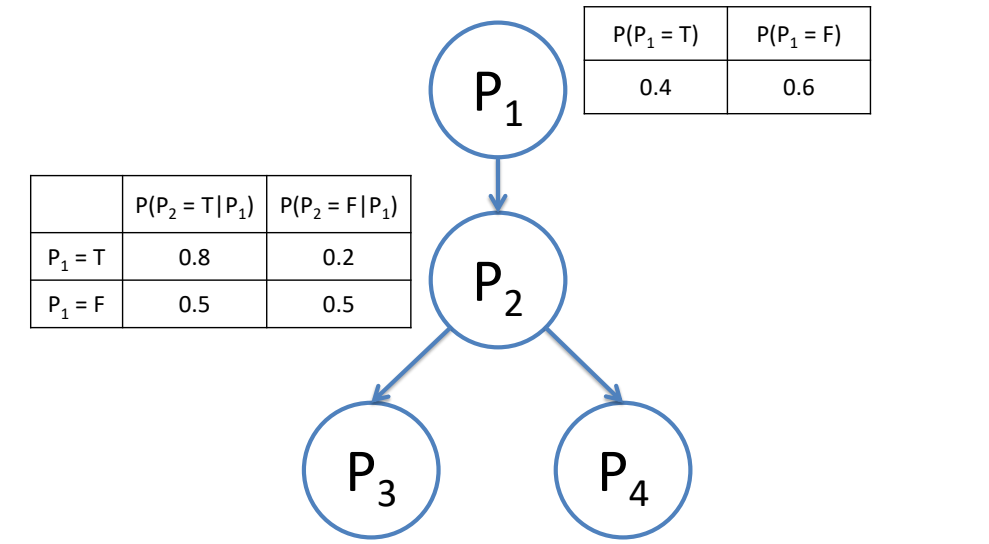
where $\exp(x) = e^x$ denotes the exponential function, and $\ln(x)$ denotes the natural logarithm. Compute the partial derivatives $\frac{\partial^2 f}{\partial x \partial y}$ and $\frac{\partial^2 f}{\partial x^2}$.

3. Does the function f defined above attain a global minimum or a global maximum? If it does, determine the points where the global extrema are attained, and determine the type of extremum for each such point. (There is more than one way to solve this part.)

13 Bayesian network (specialization UI-SU, UI-ZPJ)

Define a Bayesian network and state how such networks are constructed and used. What is the relationship between the Bayesian network and full joint probability distribution?

Consider the following Bayesian network with 4 random variables $P_1, \ldots, P_4$ and affiliated conditional probability distributions. We observed that $P_3 = F$ and $P_4 = T$. Calculate the probability of $P_1 = T$, i.e. $P(P_1 = T | P_3 = F, P_4 = T)$.



P(P1 = T)	P(P1 = F)
0.4	0.6

	P(P2 = T | P1)	P(P2 = F | P1)
P1 = T	0.8	0.2
P1 = F	0.5	0.5

	P(P3 = T | P2)	P(P3 = F | P2)
P2 = T	0.2	0.8
P2 = F	0.3	0.7

	P(P4 = T | P2)	P(P4 = F | P2)
P2 = T	0.8	0.2
P2 = F	0.5	0.5

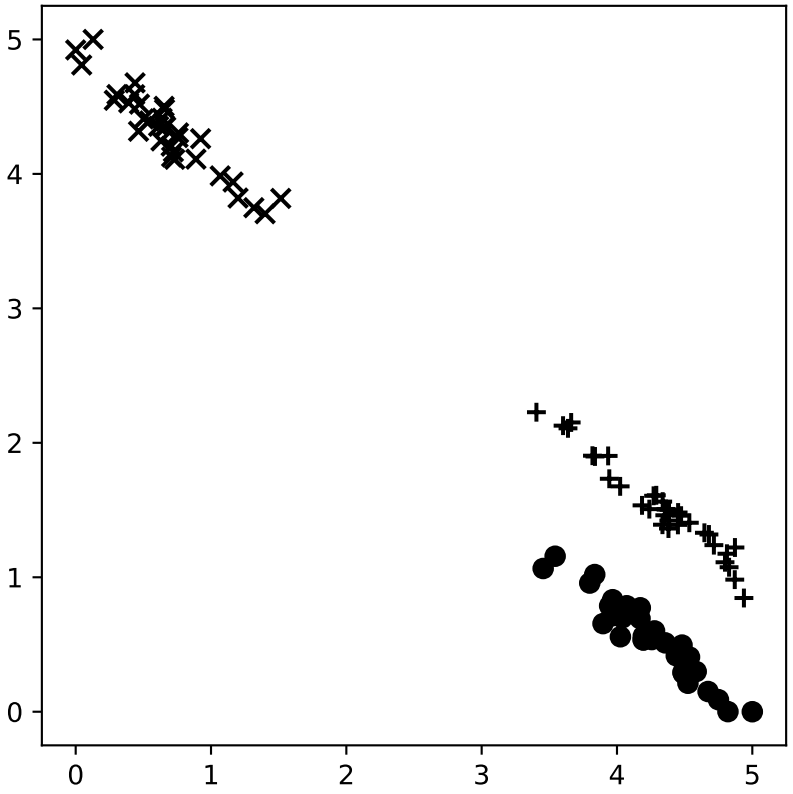
Solution sketch

- The Bayesian network is a directed acyclic graph capturing the conditional dependence of random variables represented by vertices and conditional probability distribution tables. Bayesian networks are ideal for taking an event that occurred and calculating the likelihood that any one of several possible known causes was the contributing factor. The network is created by ordering the random variables and creating the conditional probability tables such that each random variable can depend only on the previous random variables. The previous random variables which are conditionally independent are not included. Bayesian network represents the full joint probability distribution by conditional dependencies.
- Solution of the example:
  1. Express the full joint probability with the observations: $P(P_1 = T | P_3 = F, P_4 = T) = \alpha \sum_{P_2} P(P_1 = T, P_2, P_3 = F, P_4 = T)$

2. Use chain rule and conditional independence: $\alpha \sum_{P_2}(P(P_1 = T)P(P_2|P_1 = T)P(P_3 = F|P_2)P(P_4 = T|P_2))$
3. The enumeration or variable elimination algorithm may be used now.
4. The result needs to be normalized. $P(P_1 = T|P_3 = F, P_4 = T)$ a $P(P_1 = F|P_3 = F, P_4 = T)$ need to sum up to 1. To that end, the normalization constant α is used.
5. The result is aproximately $P(P_1 = T|P_3 = F, P_4 = T) = 0.4394$.

14 PCA analysis (specialization UI-SU, UI-ZPJ)

Consider data with two attributes and three classes in the figure below.



1. Explain the PCA analysis. How can we compute the principal components?
2. Without doing any calculations, estimate the directions of the two principal components for the data in the figure above. Draw them in the figure.
3. Assume the k-nearest neighbor classifier for a reasonably chosen k and assume that as a preprocessing step before its training, we reduce the dimension of the data to 1 using the PCA analysis. What will be the accuracy of the classifier compared to the same classifier trained on the original data? Why?

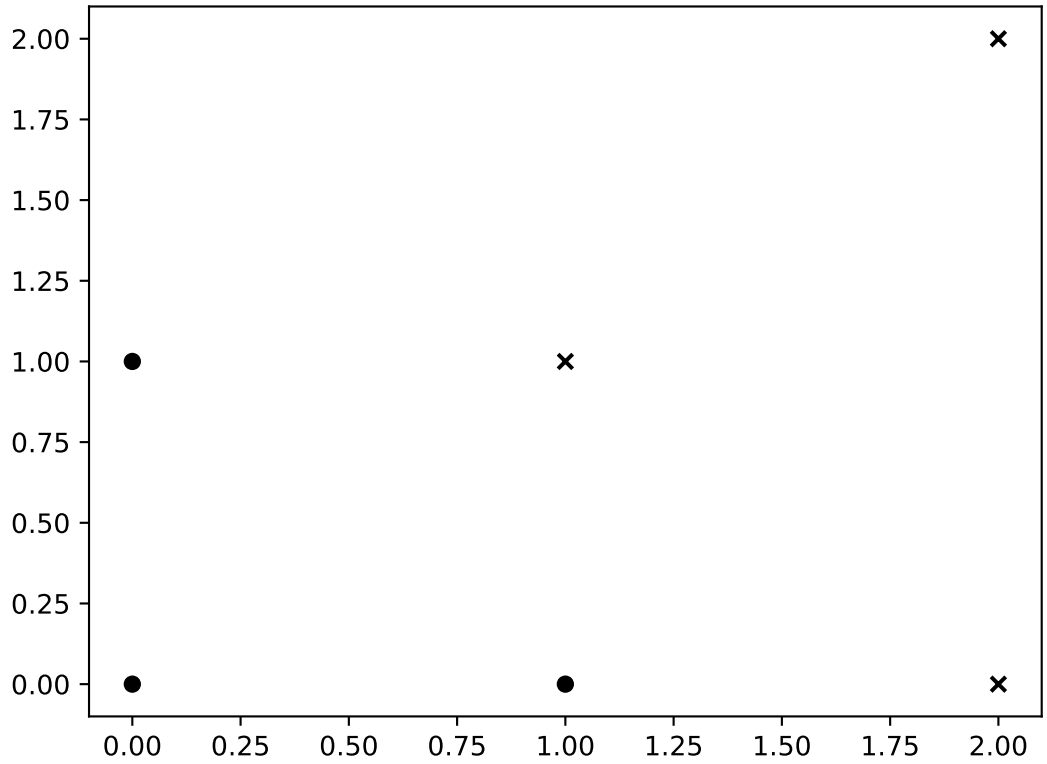
Solution sketch

1. The PCA analysis looks for directions in the data with the maximum variance. The first component w can be found by solving $\arg\max_{||w||} \sum_i (x_i \cdot w)^2$. The other components can be found by iteratively removing the previous ones from the data.

2. The first principal component will be in the direction from the top left to the bottom right corner, roughly through the middle of the data. The second principal component will be perpendicular to the first one, again, roughly in the middle of the data.
3. The projection to the first component will mix the "dot" and "+" classes making them harder to classify, so we can expect the classifier trained on the pre-processed data will have lower accuracy.

15 Support Vector Machines (SVM) (specialization UI-SU, UI-ZPJ)

Consider a two-dimensional dataset with two classes. The data points for Class A are located at (1,1), (2,2), and (2,0), and the data points for Class B are located at (0,0), (1,0), and (0,1). The data are also displayed in the figure below, instances of class A are displayed as (×) and instances of class B are displayed as (•).



1. Describe how linear Support Vector Machines work and how they are trained in case of linearly separable data.
2. Given the data above, draw an approximate decision boundary of a linear SVM that separates the two classes. You do not need to perform any calculations.
3. Without performing any calculations, explain how adding a new data point at (3,3) for Class A would affect the decision boundary if we are using a linear SVM.
4. Would using and RBF kernel improve the classification in this case? Draw a dataset, where RBF kernels would improve classification.

Solution sketch

1. Linear SVMs look for the maximum margin separating hyperplane. The answer should include the mathematical formulation of the problem (at least the primal form).

2. The separating hyperplane would be the line $y = -x + 1.5$. Any straight line being roughly the same distance from the closest instances of the two classes would be considered correct.
3. The new data point at (3,3) would not affect the existing decision boundary in a linear SVM since it is on the same side as the other Class A points and does not violate the margin.
4. The data are linearly separable, RBF kernel thus would not improved the classification in this case. RBF kernel could help if the data were not linearly separable, e.g. the case where one class is completely surrounded by another one.

16 Least squares method (specialization UI-SU)

Consider data with a single numerical attribute x_1 and a single numerical target y in the table below. We want to fit a linear model to this data using the least squares method.

x_1	y
1	1
2	3
3	2

1. Explain the linear model and the ordinary least squares method. How can this method be used to find the coefficients of the model?
2. Use the least squares method to calculate the coefficients for a linear model for the data given in the table above.
3. Calculate the mean squared error of the model.

Solution sketch

1. The linear model models the data as $Y = \beta_0 + \sum_i \beta_i x_i$. The OLS method can be used to estimate the coefficients of the model by using the formula $\beta = (X^T \cdot X)^{-1} X^T y$, where X is a matrix containing a column of all 1s (for the intercept) and then columns of values from x.
2. By using the formula above, we get $\beta_0 = 1$ a $\beta_1 = 0.5$.
3. The MSE of the model is $\frac{(1-1.5)^2+(3-2)^2+(2-2.5)^2}{3} = 0.5$.