State Final Examination (Sample Questions)

2025-09-08

1 Complexity classes (shared topics)

1. Define *complexity classes* **P** and **NP**.
2. Define a *reduction* between decision problems (languages).
3. Define **NP**-*completeness* of a decision problem.
4. Determine whether the following problem belongs to the **NP** class and justify your answer: The input is a positive integer x written in decimal. Is there a pair of positive integers a and b such that $1 < a, b < x$, $x = a \cdot b$, and decimal representations of a and b have the same number of digits (assuming no leading zeroes)?

Solution sketch For definitions, see the chapter NP-Completes in Introduction to Algorithms by Cormen et al. The given problem belongs to **NP**. As a certificate, you can use the value of a. The verifier simply computes $b = x/a$ (and rejects if division yields a non-zero remainder) and checks that $1 < a, b < x$ and the decimal representations of a and b are equally long. The length of the certificate is linear and time complexity of the verifier polynomial, both with respect to the length of the input (the number of digits of x).

2 Binary file storage (shared topics)

An application stores its data in a binary file, illustrated by the following (partial) hex dump:

```
0000:  00 06 48 68  53 53 4C 4C  00 22 03 E8  FC 18 00 05  ..Hh SSLL ."..  ....
0010:  48 65 6C 6C  6F 00 05 57  6F 72 6C 64  00 00 00 00  Hell o..W orld  ....
0020:  00 00 00 2C  00 00 00 00  00 00 12 34  00 88 01 F4  ...,  ....  ...4  ....
```

Logically, the file contains *nodes*. Each node contains the same set of *attributes*. The types of the attributes are determined by the file *header* located at the beginning of the file, consisting of two bytes containing the number of attributes (in our example, 6), followed by attribute types, encoded by one byte for each attribute, corresponding to this enumeration:

```
enum AttrType { TUInt16 = 0x48, TLink = 0x4C, TString = 0x53, TSInt16 = 0x68};
```

Multi-byte numbers stored in the file are always in the big-endian order. The *root node* is located immediately after the header. Each node starts with two bytes containing the length of the node data in bytes (in the root node of our example, 34). The node data contains the values of the attributes, encoded according to the attribute types declared in the header. For `TUInt16` and `TSInt16` types, the attribute is stored as two bytes, containing an unsigned or signed (two's complement) 16-bit integer, respectively. `TString` attributes are variable-length ASCII strings encoded as two bytes specifying the number of characters, followed by the characters, each stored in one byte. For `TLink`, there are 8 bytes containing the position of another node in the file. The partial dump shows that the root node holds the values `{1000, -1000, "Hello", "World", 0x2C, 0x1234}`.

A binary file is represented by the class `BinaryFile` that provides the following method that reads `len` bytes into the buffer `buf`, starting at absolute position `filepos` in the file (signed types are used for compatibility with java):

```
void readBytes (long filepos, byte[] buf, int len);                 // Java, C#
void readBytes (std::int64_t filepos, unsigned char* buf, int len);    // C++
```

1. Write the declaration of a class `Reader`, including its private data and the implementation of the following methods: The constructor shall receive an open `BinaryFile` object as its parameter; it shall read the header immediately. The

`attributes` method returns the number of attributes. The `getType` method returns the type of the `i`-th attribute, numbered from zero. The `getRoot` method returns the position of the root node. The `readNode` method shall return a new `Node` object representing the node stored at the position `filepos`. The object is returned by reference in Java/C# but by value in C++. The node data shall be read from the file only by this function.

2. Declare and implement the private data elements and the constructor of the `Node` class. The class shall store the node data as an array of bytes, i.e. as stored in the file. In addition, there shall be a structure that helps locate any attribute in constant time. The conversion of the bytes to integers or strings is done only when the corresponding `get...` method of the `Node` is called.

3. Write the implementation of the following methods of the class `Node` for getting the value of unsigned and signed 16-bit integer attributes, using only built-in operators of the selected language:

   ```
   int getUInt16(int i);
   int getSInt16(int i);
   ```

   Each method reads the `i`-th attribute; it is the responsibility of the caller to ensure that the call matches the type of the attribute. Note that the type `int` is considered large enough to contain the range $\langle -32768, 65535 \rangle$.

Solution sketch

```
int convertUInt16(const unsigned char* p)
{
    return ((int)p[0] << 8) + p[1];
}
// in java/C\#, the arguments should be (byte[] buf, int offs)

int convertSInt16(const unsigned char* p)
{
    return (((int)p[0] & 0x7F) << 8) + p[1] - (((int)p[0] & 0x80) << 8);
}

class Reader {
public:
    Reader(BinaryFile & bf) : f(bf)
    {
        bh = readByteArray(0);
        // check attribute types here
    }
    int attributes() const
    {
        return bh.size();
    }
    AttrType getType(int i) const
    {
        return AttrType(bh[i]);
    }
    std::int64_t getRoot() const
    {
        return 2 + bh.size();
    }
    Node readNode(std::int64_t filepos)
    {
        return Node(readByteArray(filepos), *this);
    }
private:
    BinaryFile & f;
    std::vector<unsigned char> bh;

    std::vector<unsigned char> readByteArray(std::int64_t filepos)
    {
```

```
        std::array<unsigned char, 2> bsz;
        f.readBytes(filepos, bsz.data(), 2);
        int sz = convertUInt16(bsz.data());
        std::vector<unsigned char> bdt(sz);
        f.readBytes(filepos+2, bdt.data(), sz);
        return bdt;
    }
};

class Node {
public:
    Node(std::vector<unsigned char>&&bd, const Reader & rdr)
    : bdt(std::move(bd)), offsets( rdr.attributes())
    {
        int offs = 0;
        for (int i = 0; i < rdr.attributes(); ++i)
        {
            offsets[i] = offs;
            switch(rdr.getType(i)) {
            case TUInt16: case TSInt16: offs += 2; break;
            case TLink: offs += 8; break;
            case TString: offs += 2 + convertUInt16(bdt.data() + offs); break;
            }
        }
    }

    int getUInt16(int i)
    {
        return convertUInt16(bdt.data() + offsets[i]);
    }
    // similarly for getSInt16
private:
    std::vector<unsigned char> bdt;
    std::vector<int> offsets;
};
```

3 Basic calculus (shared topics)

Let $p \in (0, +\infty)$ be a positive real number. Let R_p denote the plane region bounded from the left by the line $x = p$, from the right by the line $x = p + 1$, from the bottom by the x-axis and from the top by the graph of the function $f(x) = x + \frac{1}{x}$. In other words,

$$R_p = \left\{ (x, y) \in \mathbb{R} \times \mathbb{R};\ p \leq x \leq p + 1 \wedge 0 \leq y \leq x + \frac{1}{x} \right\}.$$

1. Show that the area of R_p is equal to $p + \ln\left(1 + \frac{1}{p}\right) + \frac{1}{2}$ square units.
2. Is there a value of p within the range $0 < p < 100$, for which the area of R_p is greater than 10 000 square units?
3. Find a value $p \in (0, +\infty)$ for which the area of R_p is minimized, or show that no such value exists.

Solution sketch

1. Let $a(p)$ denote the area of R_p. This area can be expressed by an integral as $\int_p^{p+1} f(x)dx$. The function $f(x) = x + \frac{1}{x}$

has an antiderivative $F(x) = \frac{x^2}{2} + \ln(x) + c$ for an arbitrary $c \in \mathbb{R}$. The area is thus equal to

$$\begin{aligned} a(p) &= F(p+1) - F(p) \\ &= \frac{(p+1)^2}{2} + \ln(p+1) - \frac{p^2}{2} - \ln(p) \\ &= \frac{2p+1}{2} + \ln(p+1) - \ln(p) \\ &= p + \frac{1}{2} + \ln\left(\frac{p+1}{p}\right) \\ &= p + \ln\left(1 + \frac{1}{p}\right) + \frac{1}{2}. \end{aligned}$$

2. Note that for p approaching zero from the right, the function $a(p)$ has a limit equal to $+\infty$, and in particular, $a(p)$ is not bounded from above on any right neighborhood of zero. Thus, for $p > 0$ sufficiently small, the area of R_p exceeds 10 000 square units.

3. Using basic differentiation formulas, we determine that throughout the domain $(0, +\infty)$, the derivative of $a(p)$ is

$$a'(p) = 1 + \frac{1}{1 + \frac{1}{p}} \cdot \frac{-1}{p^2} = 1 - \frac{1}{p^2 + p}.$$

By solving a quadratic equation, we find that the only positive solution to $a'(p) = 0$ is $p_0 = \frac{-1+\sqrt{5}}{2}$. Moreover, we note that $a'(p)$ is increasing throughout its domain: this can be seen directly from the formula for $a'(p)$, or by computing the second derivative

$$a''(p) = \frac{2p+1}{(p^2+p)^2},$$

which is clearly positive for $p \in (0, +\infty)$. Thus, $a'(p)$ is negative for $p \in (0, p_0)$ and positive for $p > p_0$, and hence $a(p)$ is decreasing on $(0, p_0]$ and increasing on $[p_0, +\infty)$.

We conclude that the function $a(p)$ attains its unique minimum for $p = p_0$.

4 Binary relation (shared topics)

1. State conditions that a binary relation R on a nonempty set X has to satisfy to be an equivalence. Describe it by mathematical formulas using logic symbols, quantifiers, and so on.

2. For $X = \{a, b, c, d\}$, count how many equivalences on X exist.

3. For $X = \{a, b, c, d\}$, give an example of a binary relation on X that is transitive, but is neither symmetric nor (weakly) antisymmetric. Justify that the relation satisfies the required properties.

   (Antisymmetry means $\forall x, y \in X : ((x, y) \in R \wedge (y, x) \in R) \Rightarrow x = y$.)

5 Ramsey theorem and quantity estimates (specialization OI-G-O, OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

1. For integers $n \geq k \geq 2$, consider a random coloring of the edges of a clique on n vertices with two colors; each edge is independently colored red with probability 50% and blue with probability 50%. Let m denote the number of k-element subsets M of the vertices of this clique such that all edges between the vertices of M are blue. Show that the mean value of the random variable m is less than

$$2^{k \log_2 n - \binom{k}{2}}.$$

   What does this imply for the size of the Ramsey number $R(k, k)$?

2. Let $n \geq 3$ and $k \geq 3 + 2\log_2 n$ be integers. Let $q(n,k)$ be the number of graphs on vertices $\{1,\ldots,n\}$ containing no clique of size k. Let $p(n)$ be the number of permutations of $\lfloor \frac{n^2}{10\log_2 n} \rfloor$ elements. Which of the numbers $q(n,k)$ and $p(n)$ is larger?

Solution sketch

1. For every k-element subset M, the probability that all edges between its vertices are blue is equal to $2^{-\binom{k}{2}}$. From the linearity of the expectation, we have

$$\mathbb{E}[m] = \binom{n}{k} \cdot 2^{-\binom{k}{2}} < n^k \cdot 2^{-\binom{k}{2}} = 2^{k\log_2 n - \binom{k}{2}}.$$

Ramsey's number $R(k,k)$ is the smallest integer such that in every coloring of the edges of a clique on $R(k,k)$ vertices with two colors, there exists a monochromatic clique of size k. Here, we consider not only blue-colored cliques, but also red-colored ones. The expected value of the number of monochromatic cliques of size k in a random coloring of cliques on n vertices is therefore

$$2\mathbb{E}[m] < 2^{k\log_2 n - \binom{k}{2}+1}.$$

When $n \leq 2^{k/2-1}$, this expected value is less than 1, and therefore there exists a coloring that does not contain any monochromatic clique of size k. It follows that $R(k,k) > 2^{k/2-1}$.

2. Consider a randomly chosen graph G on vertices $\{1,\ldots,n\}$. The number of cliques of size k in this random graph corresponds to the random variable m from the previous task. Using the constraint $k \geq 3 + 2\log_2 n$, we obtain that the expected value of the number of these cliques is less than

$$2^{k\log_2 n - \binom{k}{2}} \leq 2^{\frac{k(k-3)}{2} - \binom{k}{2}} = 2^{-k} < 1/2.$$

From Markov's inequality, the probability that G has at least one clique of size k is less than $1/2$. The number of all graphs with vertices $\{1,\ldots,n\}$ is $2^{\binom{n}{2}}$, and thus $q(n,k) > 2^{\binom{n}{2}-1}$.

Let $x = \lfloor \frac{n^2}{10\log_2 n} \rfloor$; clearly $x < n^2$. The number of permutations of x elements is

$$p(n) = x! \leq x^x = 2^{x\log_2 x} < 2^{2x\log_2 n} \leq 2^{\frac{n^2}{5}} < 2^{\binom{n}{2}-1} < q(n,k).$$

The number $q(n,k)$ is therefore greater than $p(n)$.

6 Optimization (specialization OI-G-O, OI-O-PADS, OI-O-PDM)

1. State the strong duality theorem for linear programming.
2. We are given an undirected graph $G = (V,E)$ and its vertices $s_1, s_2, t_1, t_2 \in V$. For $i = 1, 2$, let P_i be the set of all paths between vertices s_i and t_i in G, and let $P = P_1 \cup P_2$. Consider the following linear programme LP1 (there is a variable x_p for every $p \in P$):

$$\begin{aligned} \max \sum_{p \in P} x_p & \\ \sum_{p : e \in p} x_p \leq 1 & \quad \forall e \in E \\ x_p \geq 0 & \quad \forall p \in P \end{aligned}$$

Formulate the dual programme (use y for the vector of dual variables).

3. Consider the graph $G = (V,E)$ with $V = \{s_1, s_2, t_1, t_2, a, b, c, d\}$ and

$$E = \{s_1a,\ s_2a,\ s_1b,\ s_2b,\ ac,\ bd,\ t_1c,\ t_2c,\ t_1d,\ t_2d\}$$

(for notational simplicity, the unoriented edge between a and c is denoted by ac).

If it exists, find an optimal solution of the (primal) linear programme LP1 above, and, using the strong duality theorem, prove that the solution is optimal.

Solution sketch

1. Strong Duality Theorem for LPs: If P and D are a primal-dual pair of LPs, then one of these four cases occurs:
   (a) Both P and D are infeasible.
   (b) P is unbounded and D is infeasible.
   (c) D is unbounded and P is infeasible.
   (d) Both P and D are feasible and there exist optimal solutions x, y to P and D such that $c^T x = b^T y$.
2. The dual LP has a variable y_e for each $e \in E$, and its objective function and constraints are as follows:

$$\begin{aligned} \min \sum_{e \in E} y_e & \\ \sum_{e \in p} y_e \geq 1 & \quad \forall p \in P \\ y_e \geq 0 & \quad \forall e \in E \end{aligned}$$

3. The LP1 is a formulation of a maxflow between s_1–t_1 and s_2–t_2. The only s_1–t_1 and s_2–t_2 paths are

$$p_1 : s_1 - a - c - t_1, \quad p_2 : s_1 - b - d - t_1, \quad p_3 : s_2 - a - c - t_2, \quad p_4 : s_2 - b - d - t_2.$$

   A primal feasible solution is $x_{p_1} = 1$, $x_{p_3} = 1$, $x_{p_2} = 0$, $x_{p_4} = 0$ of objective value 2.

   A dual feasible solution is $y_{ac} = 1$, $y_{bd} = 1$, and $y_e = 0$ for all other edges; the objective value is 2, thus, by the strong duality theorem, the above primal solution (as well as the dual) is optimal.

7 Geometry (specialization OI-G-O, OI-G-PADS, OI-G-PDM)

Let B be a closed ball of radius 50 in $\mathbb{R}^3$ centered in the origin. Let $L = \mathbb{Z}^3 \cap B$ be the set of all lattice points that belong to B and let L^- be $L \setminus \{0\}$ (i. e., L minus the origin). Finally, let $\mathcal{C}$ be the collection of all the closed balls of radius $1/4$ centered in the points of L^-. Prove that there is no line passing through the origin which would avoid every ball in $\mathcal{C}$.

Hints: If there is, for contradiction, such a line ℓ, consider all the points sufficiently close to ℓ within B. Show that these points form a convex set K avoiding all lattice points (except the origin) of sufficiently large volume (or consider a slightly smaller convex set if it simplifies the computation). Use Minkowski's theorem for lattices to deduce a contradiction.

You will get partial points for a partial progress on hints (including possibly stating the Minkowski theorem in the 3-space and explaining how it could be used without checking the properties of K).

Solution sketch Úloha je 3-rozměrná varianta příkladu, co bývá běžně na přednášce. Viz Example 2.1.2 v https://kam.mff.cuni.cz/ matousek/kvg1-tb.pdf

Hint celkem přesně popisuje, co se má dělat:

Vezmeme K z hintu jako otevřený válec se středem v počátku, souměrný podle ℓ, poloměrem podstavy $1/4$ a výškou řekněme 98 (aby byl určitě uvnitř B a tedy netrefil žádný mřížový bod mimo B). Objem K je potom $\pi \times (1/4)^2 \times 98 > 2 \times (1/4)^2 \times 64 = 8$. Tedy podle Minkowského věty o mřížkách K obsahuje mřížový bod, spor.

8 Greedy graph coloring algorithm (specialization OI-G-PDM, OI-O-PDM, OI-PADS-PDM)

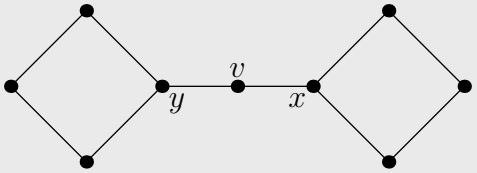
Consider the following greedy algorithm that properly colors a graph G using colors 1, 2, ...: If G has only one vertex, assign color 1 to this vertex and stop. Otherwise, select any vertex v of G of smallest degree, find a coloring of the graph $G - v$ recursively, and finally color v using the smallest color that is not assigned to any of the neighbors of v.

1. Show that this algorithm colors every planar graph G using at most 6 colors.

2. Find a bipartite graph for which this algorithm does not necessarily find a coloring using only two colors.
3. Show that this algorithm colors every chordal graph G optimally, i.e., using at most $\chi(G)$ colors.

Solution sketch

1. The minimum degree of a planar graph is at most 5 (and every induced subgraph of a planar graph is planar). Each vertex v selected by the described algorithm therefore always has degree at most 5, and when we select a color for v, at least one of the colors $1, \ldots, 6$ is not used on its neighbors. Therefore, each vertex will be assigned one of these colors.
2. Consider, for example, the following graph:



   The algorithm can first select the marked middle vertex as the vertex v of the smallest degree. During the recursive call for the subgraph $G - v$, the algorithm then finds a coloring of the two remaining 4-cycles using colors 1 and 2. Since these 4-cycles are rotationally symmetric, it may of course happen that the vertex x will receive color 1 and the vertex y will end up being colored by color 2. We will then have to use color 3 for the vertex v.
3. Since every induced subgraph of a chordal graph is chordal, similarly to planar graph case it suffices to show that every chordal graph G has minimum degree less than $\chi(G)$. Every chordal graph has a simplicial vertex u, i.e., a vertex whose neighbors form a clique. Since the chromatic number of a graph is always greater than or equal to the size $\omega(G)$ of the largest clique, we obtain $\deg u < \omega(G) \leq \chi(G)$.

9 Properties of a multivariate function (specialization OI-G-O, OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

Consider the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f(x,y) = \begin{cases} 0 & \text{for } y = 0 \\ \frac{\exp(-x^2)}{y} & \text{for } y \neq 0, \end{cases}$$

where $\exp(x)$ denotes the exponential function e^x.

1. Is the function f continuous at the point $(0,0)$?
2. Define a set $M \subseteq \mathbb{R}^2$ as follows:

$$M = \left\{ (x,y) \in \mathbb{R}^2;\ x > 0 \wedge y \geq \frac{1}{x} \right\}.$$

   Is the set M closed? Is it compact?
3. What is the maximum value that the function f attains on the set M? At which point of M is the maximum attained?

Solution sketch

1. The function f is not continuous at $(0,0)$. This can be seen for instance from the fact that the function $f(0,y) = \frac{1}{y}$ is not bounded in any punctured neighborhood of $y = 0$, and therefore has no proper limit as $y \to 0$.
2. The set M is closed (each point from its complement has a neighborhood that is contained in the complement). It is not compact, since it is not bounded.
3. The maximum of f on the set M is $\frac{e^{-1/2}}{\sqrt{2}} = \frac{1}{\sqrt{2e}}$, and is attained at the point $\left(\frac{1}{\sqrt{2}}, \sqrt{2}\right)$. To see this, note first that the partial derivatives of f are both negative on M. From this it follows that f may not attain its maximum in the interior of M: for any point (x,y) from the interior, we have $f(x,y) < f(x, \frac{1}{x})$ with $(x, \frac{1}{x})$ a point on the boundary of M.

   It is thus enough to search for maxima among the points of the form $(x, \frac{1}{x})$. The function $f(x, \frac{1}{x}) = xe^{-x^2}$ has a unique maximum, attained at $x = \frac{1}{\sqrt{2}}$, which corresponds to the sought maximum of $f(x,y)$ on M.

10 Single-move games (specialization UI-SU, UI-ZPJ, UI-ROB)

Consider one-shot games where players choose their moves simultaneously, and their payoff is determined based on the combination of chosen moves.

(A) For one-shot games of two players, write definitions of the following terms: *dominant strategy*, *Nash equilibrium*, and *Pareto optimal outcome*.

(B) In the following matrices, each cell contains two numbers representing the payoffs for players A and B after choosing the actions *Right* or *Left*, and players are maximizing the payoff. For these three one-shot games, list all dominant strategies, Nash equilibria, and Pareto optimal outcomes.

(C) Using these examples, explain the differences between all pairs of the concepts defined in part (A).

		B	
		Left	Right
A	Left	A = 10, B = 5	A = 7, B = 8
	Right	A = 0, B = 6	A = 15, B = 7

Table 1: Game (1)

		B	
		Left	Right
A	Left	A = 100, B = 100	A = 0, B = 0
	Right	A = 0, B = 0	A = 50, B = 50

Table 2: Game (2)

		B	
		Left	Right
A	Left	A = 20, B = 20	A = 0, B = 40
	Right	A = 40, B = 0	A = 10, B = 10

Table 3: Game (3)

Solution sketch

(A)
- A strategy s for player p dominates strategy s' if the outcome for s is better for p than the outcome for s', for every choice of strategies by the other player(s).
- Nash equilibrium – no player benefits by switching strategy, given that every other player sticks with the same strategy.
- Outcome is Pareto dominated by another outcome if all players would prefer the other outcome. Outcome is Pareto optimal, if there is no other outcome that all players would prefer.

(B)
- A has no dominant strategy, B has dominant strategy Right. Right, Right is Nash equilibrium. Pareto optimal outcomes are Right for B.
- There are no dominant strategies, both Left, Left and Right, Right are Nash equilibria. Pareto optimal is Left for both players.
- Both players dominant strategy is Right. Right, Right is Nash equilibrium but is not pareto optimal.

(C)
- A dominant strategy is the best response to all possible actions of the other players, while a Nash equilibrium is the best response to the actual actions of the other players. If a game has a dominant strategy equilibrium, it is also a Nash equilibrium; see (1). However, a Nash equilibrium does not necessarily require the existence of a dominant strategy; see (2).
- A dominant strategy is one that provides the best outcome for a player regardless of what other players choose, while a Pareto optimal outcome is one where no player can be made better off without making at least one other player worse off. A game can have a dominant strategy equilibrium that is not Pareto optimal; see (3). Conversely, a game can have a Pareto optimal outcome that is not a dominant strategy equilibrium; see (2).

– Nash equilibrium focuses on individual best responses, while Pareto optimality focuses on overall efficiency. They are distinct concepts; see all examples.

11 Binary classifier evaluation (specializace UI-SU, UI-ZPJ)

Imagine, your task is to select a tool for computer log analysis and detecting anomalies in them (these anomalies can be caused, for example, by hardware failure, or cyberattacks). You can select from two different tools. The available marketing materials for the first tool (Tool A) mention that it has 99.5 % accuracy and can detect 90 % of anomalies. The information about the other tool (Tool B) mentions that it can detect 81 % of anomalies and only 10 % of the detections are false positives.

Assume that both tools were evaluated on the same data containing 10,000 instances out of which only 1 % are anomalies.

1. Compute the confusion matrix of both tools and these metrics for binary classification: accuracy, precision, recall, and F1-score.
2. What are the advantages and disadvantages of the individual tools in their practical deployment? Which of them would you use? Explain. (*There is not only one correct answer.*)

Solution sketch We know that the data contain 9,900 negative instances (non-anomalies) and 100 positive instances (anomalies).

1. Tool A correctly detects 90 out of 100 anomalies, so TP = 90, and therefore FN = 10. This tool also correctly classifies 9,950 out of 10,000 instances. So it correctly classifies 9,950 - TP = 9,860 negative instances. Therefore, TN = 9,860 and FP = 40. Recall is 0.9 (given), precision is TP/(TP+FP) = 90/130 (approx. 0.69), F1-score is 0.78 and accuracy (given) is 0.995.

   Tool B correctly detects 81 out of 100 anomalies. Therefore, TP = 81, FN = 19. Only 10 % detections are false positives, therefore precision is 0.9 = TP/(TP+FN) and FN = 9. The rest of the instances (9,981) are true negatives. Recall is 0.81 (given), precision is 0.9, F1-score is 0.85 and accuracy is 0.997.
2. Tool B is better in all metrics (except recall) than Tool A. Its advantage may be that it reports much less false positive results and thus the system administrators do not have to deal with so many of them. On the other hand, if detecting the anomalies is critical, we may prefer Tool A, that can detect more of them, even though around 30 % of the detections are false positives.

12 Prediction using tabular data (specialization UI-SU, UI-ZPJ)

You are given the following agricultural dataset for crop yield prediction and based on the dataset, you want to train a machine learning model using the dataset.

Soil pH	**Rainfall (mm)**	**Temp Avg (°C)**	**Fertilizer Type**	**Seed Variety**	**Farm Size (hectares)**	**Irrigation**	**Altitude (m)**	**Yield (t/ha)**
6.8	450	22	Organic	Hybrid-A	48.6	Drip	150	4.2
7.2	380	25	Chemical	Standard	34.4	Sprinkler	200	3.8
6.5	520	18	Organic	Drought-Res	121.4	Rain-fed	850	3.2
7.0	320	28	Bio-fertilizer	Hybrid-B	60.7	Drip	75	5.1
6.9	680	20	Chemical	High-Yield	25.9	Flood	300	6.8

Based on the example dataset, answer the following questions:

1. What type of machine learning task is this?
2. Suggest an appropriate machine learning model for this task and briefly explain why it would be suitable.
3. If the suggested model requires it, describe the preprocessing steps you would apply to this dataset before training the model. Be specific about how you would handle different feature types.
4. What evaluation metric(s) would you use to assess model performance?

Solution sketch This is a *regression task* and *supervised learning* because the target variable (Yield) is a continuous numerical value and we have training data with target values.

Best model choice: *Random Forests* or *Gradient Boosting (XGBoost/LightGBM)*, good for tabular data. *Linear regression* or *MLP* might also work with careful feature preprocessing.

Preprocessing Steps:

- *Categorical features:* Apply one-hot encoding encoding for Fertilizer Type, Seed Variety, and Irrigation method.
- *Numerical features:* Check for outliers and apply standardization/normalization (e.g., StandardScaler) if using algorithms sensitive to scale. In the dataset: soil pH, rainfall, temperature, farm size, altitude.
- (optional) *Missing values:* Handle any missing data through imputation (mean/median for numerical, mode for categorical).
- (optional) *Feature engineering:* Consider creating interaction features (e.g., rainfall vs. temperature) or polynomial features if domain knowledge suggests non-linear relationships.

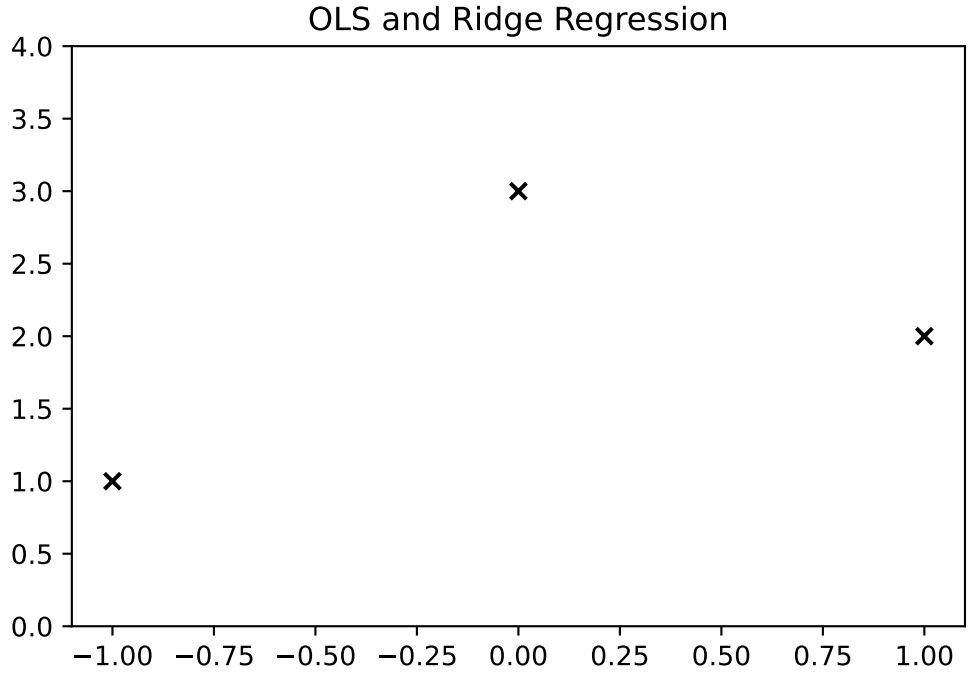
Evaluation Metrics: Mean Absolute Error, (Root) Mean Square Error,

13 Linear regression, L2 regularization (specialization UI-SU)

Consider the following training data with a single independent feature x and the target variable y. The task is to fit the linear model by the least squares method without regularization and with L2 regularization.

x	y
-1	1
0	3
1	2

1. Describe the linear model and the least squares method. How do you estimate the model coefficients?
2. Consider the training data above. Compute the linear model coefficients by the least squares method (without regularization). Draw predicted values for $x \in \langle -1, 1 \rangle$ in the graph.
3. Describe the $L2$ regularization and the target function to optimize. How will the coefficients change? Plot the prediction for $x \in \langle -1, 1 \rangle$ in the graph. Focus on the slope change compared to the estimate without regularization. You do not have to compute the values exactly.



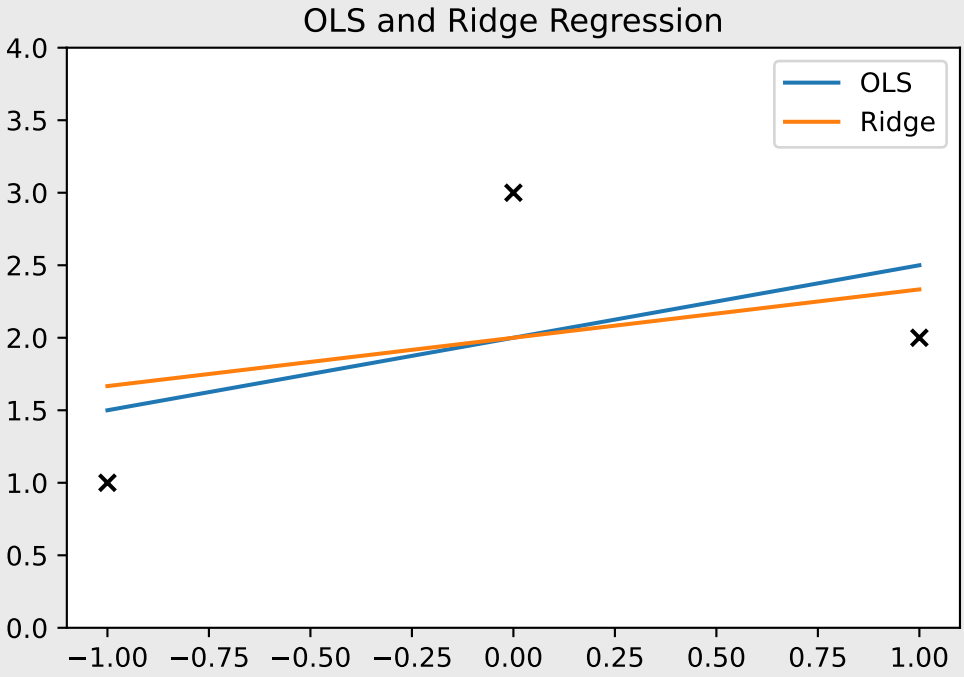
Solution sketch

1. The linear model with one feature has two coefficients β_0, β_1. The function is $f(x) = \beta_0 + \beta_1 \cdot x$. We minimize the sum over training data $\sum_{i=1,2,3}(f(x_i) - y_i)^2$. To estimate model coefficients, we add a column of 1's to the training data and solve: $\beta = (X^T X)^{-1} X^T y$.

2. The solution is:

$$\begin{aligned} \beta &= (X^T X)^{-1} X^T y \\ &= \left(\begin{bmatrix} 1,1,1 \\ -1,0,1 \end{bmatrix} \cdot \begin{bmatrix} 1,-1 \\ 1,0 \\ 1,1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1,1,1 \\ -1,0,1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = \left(\begin{bmatrix} 3,0 \\ 0,2 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 6 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{6}{3} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{1}{2} \end{bmatrix}. \end{aligned}$$

3. L2 regularization adds $\sum_{j \neq 0} \beta_j^2$ to the target function. We minimize $\sum_{i=1,2,3}(f(x_i) - y_i)^2 + \alpha\beta_1^2$, $\alpha > 0$. The coefficient β_1 is smaller than before but with the same sign. The intercept β_0 is not penalized. It does not change in this case, but it may generally change. (The R2 regularization is called Ridge in the graph).



14 TF-IDF (specialization UI-ZPJ)

1. Describe how to use TF-IDF for information retrieval. Provide the formula for computing TF-IDF, describe its components, and explain intuition why the method works.
2. Explain what data is needed for the computation, and why and how it should be preprocessed.
3. Describe how TF-IDF is used for document retrieval based on a query.

Solution sketch

1. $\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$, where $\text{TF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$ and $\text{IDF}(t) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$.
2. Combines local term importance in document (TF) with global term rarity in collection (IDF). Rare terms have higher discriminative power than frequent ones. *Extra observations:* The logarithm in IDF corrects for Zipf's law – neutralizes very frequent words. The sum of TF-IDF gives mutual information between terms in the document and the collection of documents.
3. Needed: term frequencies in documents and total number of documents. Preprocessing: tokenization, normalization (e.g., lowercase), removal of stop words.
4. For retrieval: documents and queries represented as TF-IDF vectors, similarity measured by cosine distance. Documents with highest similarity to query are returned.

15 Omnidirectional drive (specialization UI-ROB)

A robot platform is equipped with three identical "omni" wheels. The platform is symmetrical, wheel $i \in \{1,2,3\}$ position is given by distance r from the robot reference point and angle θ_i relative to the x axis, wheels' orientations are characterized by their respective unit vectors $\hat{u}_i$ perpendicular to the translation of a particular wheel i from the reference point.

Consider uniform movement of the robot on a plane. The move is characterized by translation $\vec{v_T}$ and rotational speed ω of the robot reference point.

Explain the principle of the robot motion. Draw a schematics of the robot outlining the configuration described above and the process of motion as well as the impact of that motion on each wheel. Derive the formula for speed of a wheel given specific move, i.e. determining S_i if we know $\vec{v_T}$ and ω, and the inverse formula, i.e. determining $\vec{v_T}$ and ω if we know S_1, S_2, S_3 (e.g. from odometry).

Solution sketch Movement principle: each wheel contributes to the move of the whole robot in the direction of its rotation, i.e. in the direction of its unit vector. Summing vectors of all wheels gives the actual direction of move of the reference point. For a given uniform movement this will result in either rotation on a spot (zero translational speed, non-zero rotational speed), straight move (non-zero translation, zero rotation), or circular move (both speeds non-zero), where the reference point move direction is tangent to this circle. Each wheel move is composed of two parts, one is given by its own rotation and the second (perpendicular to the rotation) is enforced by the rotation of the other wheels. Thanks to the omni wheel principle, this perpendicular part does not affect the rotation of this particular wheel (it just allows moving the wheel in that direction).

Movement formulas derivation: Move $\vec{w_i}$ of wheel i is determined by translation from the robot reference point and given movement translation and rotation. To calculate wheel speed S_i, we project this vector onto this wheel unit vector $\hat{u_i}$:

$$\vec{w_i} = \vec{v_T} + \vec{\omega} \times \vec{r_i} \tag{1}$$
$$S_i = \vec{w_i} \cdot \hat{u_i} \tag{2}$$
$$= v_{T_x} cos(\varphi_i) + v_{T_y} sin(\varphi_i) + \omega r \tag{3}$$

where vector $\vec{\omega}$ represents rotation, therefore its direction is normal to the movement plane and its size is determined by given rotational speed ω; vector $\vec{r_i}$ represents position of wheel i in respect to the reference point, unit wheel vector $\hat{u_i}$ is determined by wheel orientation angle φ_i in respect to x axis.

To get inverse formulas, we will take all three wheels speed equations (S_i given $\vec{v_T}$ and ω and platform configuration, given by wheel position and orientation $\vec{r_i}$) and resolve the simultaneous equations for $\vec{v_T}$ and ω. Since we have three equations for three variables and the equations are not dependent (thanks to given robot configuration), there will be exactly one solution.

For the given symmetrical platform, we can (without any loss) choose a configuration with angles θ_i be 0°, 120°, 240°(resp. 0, $\frac{2}{3}\pi$, $\frac{4}{3}\pi$) and therefore wheels unit vectors directions φ_i be 90°, 210°, 330°($\frac{\pi}{2}$, $\frac{7}{6}\pi$,$\frac{11}{6}\pi$). Using the constants with equation 3 we get

$$S_1 = v_{T_y} + \omega r \tag{4}$$
$$S_2 = -\frac{\sqrt{3}}{2} v_{T_x} - \frac{1}{2} v_{T_y} + \omega r \tag{5}$$
$$S_3 = \frac{\sqrt{3}}{2} v_{T_x} - \frac{1}{2} v_{T_y} + \omega r \tag{6}$$

These parallel equations will get resolved for v_{T_x}, v_{T_y}, ω:

$$v_{T_x} = -\frac{\sqrt{3}}{2} S_2 + \frac{\sqrt{3}}{2} S_3 \tag{7}$$
$$v_{T_y} = \frac{2}{3} S_1 - \frac{1}{3} S_2 - \frac{1}{3} S_3 \tag{8}$$
$$\omega = \frac{1}{3r}(S_1 + S_2 + S_3) \tag{9}$$

16 Denavit-Hartenberg system (specialization UI-ROB)

Describe the Denavit-Hartenberg system for kinematic chain description and derive transformation matrices for all substeps of a single chain step. Derive also the resulting single step transformation matrix.

Outline the system construction for a generic kinematic chain consisting of rotational and translational joints. Describe the general properties of the resulting D-H table considering constants and variables of the steps.

Solution sketch See `https://w.wiki/EJDQ`.

17 Image segmentation (specializace UI-ROB)

This question is about image segmentation.

1. Define image segmentation. The definition contains five properties of the segmentation. Give an example of a logical predicate that can be used for segmentation.
2. Describe thresholding techniques for image segmentation. Describe the Otsu's thresholding in detail.
3. Describe the steps of the K-means algorithm. Describe one method to select the K parameter. How can we use K-means for image segmentation?

Solution sketch

1. Define image segmenation. The definition contains five properties of the segmentation. Give an example of a logical predicate that can be used for segmentation.

   Definition Let R denote the spatial region of the whole image. Then segmentation is the division of R into n regions $R_1, R_2, \ldots, R_n$ such that:

   (a) $\bigcup_{i=1}^{n} R_i = R$

   (b) R_i is a connected set, $\forall i = 1, 2, \ldots, n$

   (c) $R_i \cap R_j = \emptyset, \forall i, j; i \neq j$

   (d) $Q(R_i) = \text{TRUE}, \forall i = 1, 2, \ldots, n$

   (e) $Q(R_i \cup R_j) = \text{FALSE}$ for adjacent R_i, R_j,

   where $Q(R_k)$ is a logical predicate defined over points of R_k, and the regions R_i and R_j are adjacent if $R_i \cup R_j$ is connected.

   Example Predicate

   $$|f(i,j) - f(\tilde{i},\tilde{j})| \leq t, \quad (i,j) \in R_k, (\tilde{i},\tilde{j}) \in R_k$$

2. Describe thresholding techniques for image segmentation. Describe the Otsu's thresholding in detail.

   Thresholding is the simplest technique, computationally inexpensive and fast

   Thresholding maps an input image $f(i,j)$ to the output image $g(i,j)$ such that:

   $$g(i,j) = \begin{cases} 1, & \text{if } f(i,j) \geq T \text{object} \\ 0, & \text{if } f(i,j) < T \text{ background} \end{cases}$$

   The image variance is defined as a constant for each threshold:

   $$\sigma_i^2 = \sigma_b^2(t) + \sigma_w^2(t)$$

   Intra-class (within) Variance – Minimize

   $$\sigma_w^2(t) = P_0(t)\sigma_0^2(t) + P_1(t)\sigma_1^2(t)$$

   Inter-class (between) Variance – Maximize

   $$\sigma_b^2(t) = P_0(t)(\mu_0(t) - \mu_I)^2 + P_1(t)(\mu_1(t) - \mu_I)^2 = P_0(t)P_1(t)(\mu_0(t) - \mu_1(t))^2$$

3. Describe the steps of the K-means algorithm. Describe one method to select the K parameter. How can we use K-means for image segmentation?

K-Means Algorithm

(a) Randomly distribute the initial means (K).

(b) Determine the assignment C for the given means:

$$C(i) = \underset{1 \leq k \leq K}{\arg\min} \|x_i - m_k\|^2, \quad i = 1, \ldots, N$$

(c) For a given assignment C, calculate the means m_k:

$$m_k = \frac{\sum_{x_i \in C_k} x_i}{N_k}$$

(d) Repeat steps 2 and 3 until stopping criteria are met:

- Mean Squared Error (MSE) $<$ threshold, or
- No change in means.

Properties

- Will converge
- Not necessarily to the global optimum
- Sensitive to noise and outliers
- Sensitive to initial mean
- Convex clusters

K parameter Compute WCSS for different K, find Inflection (elbow) point

K-Means for segmentation Feature space: Brightness / color / texture + position (to have connected components)