

Kapitola 1

Základne pojmy

1.1 Základy matematickej logiky

Výrok je gramatická veta, ktorá je pravdivá, alebo nepravdivá. O jej pravdivosti vieme jednoznačne rozhodnúť. Výrok musí byť oznamovacia veta.

Každému výroku vieme priradiť jeho pravdivostnú hodnotu. Ak je výrok pravdivý, priradíme mu hodnotu 1 a ak je nepravdivý, hodnotu 0. Z výrokov vieme vytvárať nové výroky pomocou logických spojok medzi ktoré patrí negácia, konjunkcia, disjunkcia, implikácia a ekvivalencia. Takto vytvorený výrok nazývame **zložený výrok**.

Nech A, B sú výroky, potom pre označenie logických spojok používame nasledujúce označenia:

1. NEGÁCIA: $\neg A, \bar{A}, A'$ (Čítame: Nie je pravda, že A .) Negovaný výrok nadobúda opačnú pravdivostnú hodnotu, ako výrok pôvodný.
2. KONJUNKCIA: $A \wedge B$ (Čítame: A a B .)
3. DISJUNKCIA: $A \vee B$ (Čítame: A alebo B .)
4. IMPLIKÁCIA: $A \Rightarrow B$ (Čítame: Ak A , tak B . resp. Keď A , potom B .)
5. EKVIVALENCIA: $A \Leftrightarrow B$ (Čítame: A práve vtedy, keď B . resp. A vtedy a len vtedy, keď B).

Pri tvorení nových matematických výrokov je dôležité upozorniť na dva fakty:

- V hovorovej reči chápeme spojku alebo vo vylučovacom zmysle, no v matematike sa chápe disjunkcia v nevylučovacom zmysle. Napr. vetu - „Zajtra o 13:00 budem doma, alebo v práci.” - chápeme tak, že môže nastať len práve jedna z možností. Nemôžeme byť na oboch miestach súčasne. Jedna sa teda o

výrok pravdivý, ak platí práve jeden z elementárnych výrokov. V matematike sa však spojka alebo chápe v nevylučovacom zmysle. To znamená, že predpokladáme, že môžu nastať aj oba elementárne výroky súčasne. Disjunkciu v matematike teda považujeme za pravdivú, ak je aspoň jeden z elementárnych výrokov pravdivý.

- Matematika chápe implikáciu doslova. To znamená, že sa jedná o výrok pravdivý v prípade, ak:
 - platia oba výroky A aj B súčasne, alebo
 - ak výrok A neplatí. V tomto prípade hovoríme, že implikácia platí triviálne.

Ako príklad si vezmeme vetu - „ Ak bude zajtra pekné počasie, pôjdem na túru.” - O situácii, keď zajtra pekne počasie nebude, nehovoríme nič. Ak nastala situácia, že pekné počasie nie je, triviálne predpokladáme, že sme neklamali a výrok je pravdivý.

Nech A, B sú výroky a nech $A \Rightarrow B$. Pre symbol implikácie vieme použiť niekoľko výrazov. Daný výrok môžeme prečítať nasledovne:

1. A implikuje B , alebo
2. B vyplýva z A , alebo
3. A je postačujúca podmienka pre výrok B , alebo
4. B je nutná podmienka pre A .

Výrok, ktorý je vždy pravdivý nazývame **tautológia**. Výrok, ktorý je vždy nepravdivý nazývame **kontradikcia**. Ak existuje aspoň jedna možnosť, kedy je výrok pravdivý hovoríme, že výrok je **splniteľný**.

Ďalšou možnosťou ako vytvárať nové výroky sú kvantifikátory. Poznáme dva základne kvantifikátory:

1. Symbol $\exists$ nazývame existenčný kvantifikátor (čítame: existuje).
2. Symbol $\forall$ nazývame všeobecný kvantifikátor (čítame: pre všetky).

V práci ďalej budeme používať aj špeciálny typ existenčného kvantifikátora. Jedná sa o symbol $\exists!$, ktorý čítame ako existuje práve jeden. Poznamenajme, že negáciou existenčného kvantifikátora získame kvantifikátor všeobecný a naopak.

Nech A, B a nech $A \Rightarrow B$ je implikácia dvoch elementárnych výrokov. Nazvime vetu $A \Rightarrow B$, veta **všeobecná**. Potom vetu:

1. $B' \Rightarrow A'$ nazývame veta obmenená.
2. $B \Rightarrow A$ nazývame veta obrátena.
3. $A \wedge B'$ nazývame negácia vety.

Veta všeobecná a veta obmenená nadobúdajú rovnaké pravdivostné hodnoty.

Definícia 1.1 *Nech A je výrok a nech množina M je neprázdnu množinou výrokov. Množinu výrokov M nazývame splniteľná práve vtedy, keď existuje také ohodnotenie výrokov, pri ktorom je pravdivý každý výrok z množiny M . Ak množina výrokov M nie je splniteľná, nazýva sa nesplniteľná. Hovoríme, že výrok A vyplýva z množiny výrokov M (A je sémantickým dôsledkom M) práve vtedy, keď A je pravdivý pri každom ohodnotení výrokových premenných, pri ktorom je pravdivý každý výrok z množiny M .*

1.1.1 Úlohy

1. Rozhodnite, ktoré z nasledujúcich viet sú výroky. Ak sa jedná o výrok, rozhodnite, či je pravdivý, alebo nepravdivý.
 - (a) Číslo 3 je prvočíslo.
 - (b) Číslo 20 je nepárne.
 - (c) Dnes je piatok.
 - (d) Počítaj!
 - (e) $\frac{0}{0} = 0$.
 - (f) Modré auto.
 - (g) $x > 2$.
 - (h) Zajtra bude pekne.
 - (i) V štvoruholníku sa uhlopriečky rozpolujú.
 - (j) $2x + 5 = 0$.
 - (k) Vonku prší.
 - (l) Množina všetkých bodov rovnako vzdialených od dvoch pevných bodov A, B je os úsečky AB .
2. Utvorte negácie nasledujúcich výrokov:
 - (a) Číslo 3 je prvočíslo.
 - (b) $3 < 5$.
 - (c) $7 = 2$.

- (d) Nie všetky čísla sú kladné.
 - (e) Číslo 75 je deliteľné číslom 15.
 - (f) Vltava je dlhšia ako Dunaj.
3. Vyjadrite symbolmi logiky. Pre vetu - Príde matka. - použite symbol M a pre vetu - Príde otec. - použite symbol O .
- (a) Ak príde otec, tak nepríde matka.
 - (b) Príde aspoň jeden z rodičov.
 - (c) Príde najviac jeden z rodičov.
 - (d) Príde práve jeden z rodičov.
4. Dané sú výroky: Kreslím (K). Píšem (P). Tancujem (T). Symbolicky zapíšte nasledujúce zložené výroky:
- (a) Keď spievam, nekreslím.
 - (b) Nepíšem, alebo netancujem, práve vtedy, keď kreslím.
 - (c) Netancujem, píšem.
 - (d) Ak nespievam a nekreslím, píšem.
5. Nech A, B sú výroky. A : Číslo končí nulou. B : Číslo je párne. Utvorte výroky:
- (a) $A \Rightarrow B$.
 - (b) $A' \Rightarrow B'$.
 - (c) $B \Rightarrow A$.
 - (d) $B' \Rightarrow A'$.
- a rozhodnite, ktorý výrok je pravdivý a ktorý je nepravdivý.
6. Jožo povedal: „ Ak dopoludnia napíšem dobre písomku z matematiky, večer pôjdem do kina. ” Sú štyri možnosti:
- (a) Jožo napísal písomku dobre, večer šiel do kina.
 - (b) Jožo napísal písomku dobre, večer nešiel do kina.
 - (c) Jožo nenapísal písomku dobre, večer šiel do kina.
 - (d) Jožo nenapísal písomku dobre, večer nešiel do kina.

V ktorých prípadoch dodržal Jožo slovo?

7. V dielni sú tri stroje, ktoré pracujú na základe týchto troch podmienok:

- (a) Ak pracuje prvý stroj, pracuje aj druhý stroj.
- (b) Pracuje druhý, alebo tretí stroj.
- (c) Ak nepracuje prvý stroj, nepracuje ani tretí stroj.

Rozhodnite, aké sú možnosti pre prácu týchto troch strojov.

8. Zistíte pravdivosť zloženého výroku: „Dnes je pondelok, alebo sa hrá futbal.”, ak v čase vypovedania výroku :

- (a) je pondelok aj sa hrá futbal.
- (b) je pondelok a futbal sa nehraje.
- (c) nie je pondelok a futbal sa hrá.
- (d) je streda a futbal sa nehraje.

9. Nech M, B, P, R označujú výroky: Murujem (M). Betónujem (B). Polievam (P). Rozmýšľam (R). Symbolicky zapíšte nasledujúce zložené výroky:

- (a) Keď murujem, nepolievam.
- (b) Nepolievam práve vtedy, keď murujem alebo betónujem.
- (c) Nebetónujem, murujem.
- (d) Ak rozmýšľam a polievam, tak betónujem.

10. Dvaja študenti čakajú na troch spolužiakov (Ivan, Peter a Monika), ktorí sú na skúške. Prvý hovorí: „ Ak skúšku urobí Ivan a Peter, určite ju urobí aj Monika.” Druhý hovorí: „ Ja si zase myslím, že ak skúšku urobí Ivan a Monika nie, tak Peter skúšku nespraví.” Tvrdia obaja to isté?

11. Olinka tvrdí: „ Keď pijem kávu, dám si aj čokoládu. Ak nejem koláč, nejem ani čokoládu.” Vyplýva z uvedených tvrdení, že:

- (a) Ak Olinka je koláč, tak nepije kávu?
- (b) Ak pije kávu, tak je aj koláč?

12. Tri obchody sú otvorené podľa týchto pravidiel: Otvorený je prvý obchod, alebo druhý nie je otvorený. Keď nie je otvorený tretí, prvý tiež nie je otvorený, ale druhý je. Ak nie je pravda, že je otvorený prvý, alebo tretí, tak je určite otvorený druhý. Rozhodnite, ktorý z nasledujúcich výrokov vyplýva z uvedených pravidiel.

- (a) Ak je tretí obchod otvorený a druhý nie, tak prvý je otvorený.
- (b) Ak je otvorený druhý obchod, sú otvorené aj zvyšné dva.

- (c) Ak nie je otvorený prvý, ani druhý obchod, tak je otvorený obchod tretí.
13. Určte predpoklady a tvrdenia nasledujúcich viet:
- (a) Súčin dvoch po sebe nasledujúcich párnych čísel je deliteľný číslom 8.
 - (b) Keď v trojuholníku je jeden uhol pravý, ostatné dva sú ostré.
 - (c) Pre pravouhlý trojuholník so stranami a, b, c platí: $a^2 + b^2 = c^2$.
 - (d) V trojuholníku je každá strana menšia ako súčet a väčšia ako rozdiel ostatných dvoch strán.
14. Zistíte, ktoré z predpokladov sú nutnými a ktoré postačujúcimi podmienkami:
- (a) Súčin dvoch čísel sa rovná nule, ak aspoň jeden z činiteľov sa rovná nule.
 - (b) Ak celé číslo končí číslicou 5, tak celé číslo je deliteľné číslom 5.
15. Napíšte v symbolickej forme (pomocou znakov logických spojok):
- (a) Nie len A , ale aj B .
 - (b) A a nie B .
 - (c) Hoci nie A , predsa B .
 - (d) Ani A , ani B .
 - (e) Nie je pravda, že A a súčasne aj B .
 - (f) Alebo nie A , alebo nie B .
 - (g) Nie je pravda, že A alebo B .
 - (h) Alebo nie A , alebo B .
 - (i) B predpokladajúc, že A .
 - (j) Nie A , ak nie B .
 - (k) Ak A , tak nie B .
 - (l) Nie je pravda, že ak A , tak aj B .
 - (m) Ak A , tak B , ale ak nie A , tak nie B .
 - (n) Ak A , tak nie B , ale ak nie A , tak B .
16. Utvorte negácie nasledujúcich výrokov:
- (a) Niektoré ženy, majú ryšavé vlasy.
 - (b) Všetky banány sú žlté.

- (c) Každý študent sa usilovne učí.
- (d) V každom ročníku FEI je študent, ktorý je usilovný.
- (e) V našej knižnici je kniha, ktorá má na každej strane tlačovú chybu.
- (f) Všetky školy v našom okrese majú nekvalifikovaného učiteľa.
- (g) V niektorej rodine sú všetci jej členovia vodiči.
- (h) Každému sa niekedy niečo nepáči.
- (i) Všetky školy v našom okrese majú triedu, v ktorej sú všetci študenti usilovní.
- (j) Existuje knižnica, ktorej všetky knihy majú na niektorej strane tlačovú chybu.
- (k) Je nemocnica, v ktorej každá sestrička ešte nejakého pacienta nevidela.
- (l) Existuje galaxia, v ktorej má každá slnečná sústava planétu, ktorej všetky satelity sú väčšie ako náš mesiac.
- (m) V niektorých knižniciach sú v každom oddelení nejaké knihy, ktoré majú na každej strane tlačovú chybu.

17. Utvorte negácie nasledujúcich výrokov:

- (a) Daná rovnica má aspoň dva reálne korene.
- (b) Najviac štyria žiaci získali vyznamenania.
- (c) Aspoň jeden žiak rieši olympiádu.
- (d) V tejto záhrade je aspoň dvadsať jabloní.
- (e) Každý deň je dôvod na radosť.
- (f) Nebol ani doma, ani v práci.
- (g) Dnes stihnem aj prácu, aj zábavu.
- (h) Bol som tam aspoň raz.
- (i) Každé celé číslo je súčinom dvoch prvočísel.
- (j) Nieкто z nás tam nebol.
- (k) Nikdy nikto nič nevie.
- (l) Každý trojuholník má aspoň jednu vpísanú kružnicu.
- (m) Každý trojuholník má práve jednu vpísanú kružnicu.

18. Utvorte negácie nasledujúcich výrokov:

- (a) $(\exists a \in \mathbb{R}) (a^2 + 1 < 0)$,

- (b) $(\forall x \in \mathbb{C}) ((x \in A) \vee (x \in B)),$
- (c) $(\exists x \in \mathbb{C}) ((x \in A) \Rightarrow (x \in B)),$
- (d) $(\forall x \in \mathbb{C}) ((x \in A) \wedge (x \in B)),$
- (e) $(\exists x \in \mathbb{C}) ((x \in A) \Leftrightarrow (x \in B)),$
- (f) $(\forall x \in \mathbb{Z}) (\forall y \in \mathbb{Z}) (\forall z \in \mathbb{Z}) (x + y = z),$
- (g) $(\forall x \in \mathbb{R}) (\forall y \in \mathbb{R}) (\exists z \in \mathbb{R}) (\frac{x}{y} = z),$
- (h) $(\forall \varepsilon \in \mathbb{R}^+) (\exists \delta \in \mathbb{R}^+) (\forall x \in (a - \delta, a + \delta)) (|f(x) - f(a)| < \varepsilon),$
- (i) $(\forall \varepsilon \in \mathbb{R}^+) (\exists x_0 \in \mathbb{R}) (\forall x \geq x_0) (|f(x) - fb| < \varepsilon).$

1.2 Matematický dôkaz

Axióma je výrok, ktorý považujeme za pravdivý. Nedokazujeme ho.

Platnosť (pravdivosť) matematických viet sa overuje (dokazuje) pomocou definícií, axióm a už dokázaných viet. Budeme rozlišovať 5 typov matematického dôkazu a to:

1. dôkaz priamo,
2. dôkaz nepriamo,
3. dôkaz sporom,
4. dôkaz matematickou indukciou,
5. dôkaz rozborom všetkých možností.

Dôkaz priamo: Nech $A \Rightarrow B$ je výrok. Vytvárame postupnosť výrokov $A_1, A_2, A_3, \dots, A_n$, kde $A \Rightarrow A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \dots \Rightarrow A_n \Rightarrow B$. Výroky $A_i \Rightarrow A_{i+1}$, pre $i \in \{1, 2, 3, \dots, n-1\}$ sú buď axiómy, definície, alebo už dokázané vety. Ak veta nie je v tvare implikácie, často ju vieme na tvar implikácie preformulovať. Napríklad vetu - Druhá mocnina prirodzeného, nepárneho čísla je prirodzené, nepárne číslo. - vieme preformulovať na vetu - Ak $n \in \mathbb{N}$ je nepárne, tak $n^2 \in \mathbb{N}$ a je nepárne.

Príklad 1.1 Dokážte nasledujúce tvrdenie priamo: Druhá mocnina prirodzeného, nepárneho čísla je prirodzené, nepárne číslo.

Danú vetu preformulujeme tak, aby sme dostali tvrdenie v tvare implikácie. Dostávame tvrdenie: $(\forall n \in \mathbb{N}) : n \text{ je nepárne} \Rightarrow n^2 \text{ je nepárne}$. Začneme vytvárať postupnosť výrokov:

- $A = A(0) : n \in \mathbb{N} \text{ je nepárne} \Leftrightarrow$
- $A(1) : n = 2k + 1, k \in \mathbb{N} \cup \{0\} \Leftrightarrow$
- $A(2) : n^2 = (2k + 1)^2, k \in \mathbb{N} \cup \{0\} \Leftrightarrow$
- $A(3) : n^2 = 4k^2 + 4k + 1, k \in \mathbb{N} \cup \{0\} \Leftrightarrow$
- $A(4) : n^2 = 2(2k^2 + 2k) + 1, k \in \mathbb{N} \cup \{0\} \Leftrightarrow$
- $A(5) : n^2 = 2l + 1, l = 2k^2 + 2k \in \mathbb{N} \cup \{0\} \Leftrightarrow$
- $A(6) : n^2 \text{ je nepárne číslo} = B$

Dôkaz nepriamo: Nech $A \Rightarrow B$ je výrok. Z výroku vytvoríme obmenenú vetu v tvare $B' \Rightarrow A'$. Túto vetu dokazujeme priamo. Využijeme fakt, že pravdivostné hodnoty oboch viet sú rovnaké.

Príklad 1.2 *Dokážte nasledujúce tvrdenie nepriamo: Ak n^2 je prirodzené, párne číslo, tak aj n je prirodzené, párne číslo.*

Najprv si vytvoríme matematický zápis daného tvrdenia: $(\forall n \in \mathbb{N}) : n^2 \text{ je párne} \Rightarrow n \text{ je párne}$. Následne z daného tvrdenia vytvoríme obmenenú vetu: $(\forall n \in \mathbb{N}) : n \text{ je nepárne} \Rightarrow n^2 \text{ je nepárne}$. Dostávame rovnaké tvrdenie ako v predchádzajúcom príklade, ktoré dokazujeme rovnako - priamo.

Dôkaz sporom: Nech $A \Rightarrow B$ je výrok. Z výroku vytvoríme negovaný výrok v tvare $A \wedge B'$. Tento výrok dokazujeme priamo. Na rozdiel od dôkazu priamo, na konci dôjdeme k sporu, teda k tvrdeniu o ktorom vieme rozhodnúť, že určite nie je pravdivé. To znamená, že dokazovaná negácia vety neplatí a teda musí platiť pôvodná veta.

Príklad 1.3 *Dokážte nasledujúce tvrdenie sporom: Druhá mocnina párneho čísla je číslo párne.*

Najprv si vytvoríme matematický zápis daného tvrdenia: $(\forall n \in \mathbb{N}) : n \text{ je párne} \Rightarrow n^2 \text{ je párne}$. Následne z daného tvrdenia vytvoríme negáciu vety: $(\exists n \in \mathbb{N}) : n \text{ je párne} \wedge n^2 \text{ je nepárne}$. Toto tvrdenie dokazujeme priamo. T.j. vytvárame postupnosť výrokov:

- $A = A(0) : n \in \mathbb{N} \text{ je párne} \Leftrightarrow$
- $A(1) : n = 2k, k \in \mathbb{N} \Leftrightarrow$
- $A(2) : n^2 = (2k)^2, k \in \mathbb{N} \Leftrightarrow$
- $A(3) : n^2 = 4k^2, k \in \mathbb{N} \Leftrightarrow$
- $A(4) : n^2 = 2(2k^2), k \in \mathbb{N} \Leftrightarrow$
- $A(5) : n^2 = 2l, l = 2k^2 \in \mathbb{N} \Leftrightarrow$
- $A(6) : n^2 \text{ je párne číslo} = B'$, čo je v spore s dokazovanou vetou. Negácia vety neplatí, platí pôvodne tvrdenie.

Dôkaz matematickou indukciou (MI) je metóda dokazovania matematických viet a tvrdení, ktorá sa používa, ak chceme ukázať, že dané tvrdenie platí pre všetky prirodzené čísla, prípadne inú, dopredu danú nekonečnú postupnosť. Predpokladajme, že chceme dokázať platnosť výroku $A(n)$ pre každé $n \in \mathbb{N}$ také, že $n \geq n_0$. Dôkaz matematickou indukciou pozostáva z dvoch častí:

1. Dokážeme platnosť tvrdenia pre $n = 1$, resp. pre najmenší prvok z uvažovanej množiny $n = n_0$. Toto nazývame báza indukcie. Z prvého kroku vyplýva, že existuje aspoň jedno také $n = n_0$, pre ktoré tvrdenie platí.
2. Ak existuje aspoň jedno také n , môžeme predpokladať, že tvrdenie $A(n)$ platí pre $\forall n \leq k$. Toto nazývame indukčný predpoklad (IP).
3. Ukážeme, že z platnosti tvrdenia pre $n = k$ (IP) vyplýva platnosť tvrdenia pre $n = k + 1$. Na túto časť dokazovania môžeme použiť už iné typy dôkazov.

T.j. Pre každé $n \in \mathbb{N}$ platí: z indukčného predpokladu - platnosti $A(n)$ dokážeme, že platí aj $A(n + 1)$. Tým je tvrdenie $A(n)$ dokázané. Dostávame: Ak platí $A(1)$ tak platí aj $A(2)$, ak platí $A(2)$ tak platí aj $A(3)$, ..., a teda $A(n)$ platí $\forall n \in \mathbb{N}$.

Príklad 1.4 Dokážte nasledujúce tvrdenie matematickou indukciou: Súčet prvých n po sebe idúcich prirodzených čísel je rovný číslu $\frac{n(n+1)}{2}$.

1. Nech $n = 1$, potom $1 = \frac{1(1+1)}{2} = 1 \dots$ tvrdenie platí pre najmenšie n .
2. Predpokladáme, že veta platí pre všetky $n, n \leq k$. Je potrebné dokázať, že platí aj pre $n = k + 1$:

$$1+2+\dots+k+(k+1) = \frac{k(k+1)}{2} + (k+1) = \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2},$$

čo je tvrdenie $V(k+1)$.

Dôkaz rozborom všetkých možností: Tento spôsob dôkazu je častokrát zdĺhavý, komplikovaný a u matematikov nie veľmi obľúbený.

Poznamenajme, že ak je veta v tvare implikácie $A \Leftrightarrow B$, tak ju dokazujeme v dvoch častiach: $A \Rightarrow B$ a zároveň $B \Rightarrow A$.

1.2.1 Úlohy

1. K nasledujúcim vetám napíšte obmenené, obrátené vety a negácie.
 - (a) Ak je trojuholník ABC pravouhlý, tak preň platí Pytagorova veta.
 - (b) Pre každé celé číslo x platí: Ak je číslo x deliteľné deviatimi, tak je deliteľné tromi.
 - (c) $(\forall x \in \mathbb{R}) (\forall y \in \mathbb{R}) : 2|x \wedge 2|y \Rightarrow 2|(x+y)$
 - (d) $(\forall x \in \mathbb{R}) (\forall y \in \mathbb{R}) : (x > 0 \wedge y > 0) \Rightarrow xy > 0$
 - (e) $(\forall x \in \mathbb{R}) (\forall y \in \mathbb{R}) : ((x > 0 \wedge y > 0) \vee (x < 0 \wedge y < 0)) \Rightarrow xy > 0$

- (f) $(\forall x \in \mathbb{R}) (\forall y \in \mathbb{R}) : (xy = 0 \wedge x \neq 0) \Rightarrow y = 0$
- (g) Ak súčin dvoch čísel je párny, tak nie sú obidve čísla nepárne.
- (h) Ak funkcia f má v každom bode intervalu (a, b) kladnú deriváciu, tak funkcia f je na intervale (a, b) rastúca.

2. Dokážte nasledujúce tvrdenia:

- (a) $\sqrt{7 + \sqrt{7}} \geq 1 + \sqrt{7 - \sqrt{7}}$
- (b) $\sqrt{11 + \sqrt{10}} < 1 + \sqrt{11 - \sqrt{10}}$
- (c) $\sqrt{\sqrt{8} + \sqrt{6}} < 2 + \sqrt{\sqrt{8} - \sqrt{6}}$
- (d) $(\forall x \in \mathbb{R}) : \frac{x^2}{1+x^4} \leq \frac{1}{2}$
- (e) $(\forall x, y \in \mathbb{R}) : -1 < \frac{x^2-1}{x^2+1} < 1$
- (f) Súčin dvoch nepárnych čísel je nepárne číslo.
- (g) Súčet dvoch párných čísel je párne číslo.
- (h) Súčet dvoch nepárnych čísel je párne číslo.
- (i) Súčet párneho a nepárneho čísla je nepárne číslo.
- (j) Súčin dvoch po sebe idúcich prirodzených čísel je vždy deliteľný dvoma.
- (k) Súčin dvoch ľubovoľných párných čísel je deliteľný štyrmi.
- (l) Ak $x \notin A \cup B \Rightarrow x \notin A \wedge x \notin B$.
- (m) Ak $x \notin A \cap B \Rightarrow x \notin A \vee x \notin B$.
- (n) Kvadratická rovnica $ax^2 + bx + c = 0$, kde $a \neq 0$ má aspoň jeden reálny koreň rovný 0 práve vtedy, keď $c = 0$.
- (o) $(\forall x, y \in \mathbb{R}) : x^2 + y^2 \geq 2xy$.
- (p) Súčet kladného reálneho čísla a jeho prevrátenej hodnoty nie je menší ako 2. Je možné, aby sa rovnal dvom?
- (q) Ak celé číslo x nie je deliteľné tromi, tak $x^2 - 1$ je deliteľné tromi.

3. Pomocou matematickej indukcie dokážte nasledujúce tvrdenia pre $\forall n \in \mathbb{R}$:

- (a) $1 + 3 + 5 + \dots + (2n - 1) = n^2$.
- (b) $1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{1}{6}n(n+1)(2n+1)$.
- (c) $1^2 + 3^2 + 5^2 + \dots + (2n-1)^2 = \frac{1}{3}n(2n-1)(2n+1)$.
- (d) $1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + n(n+1) = \frac{1}{3}n(n+1)(n+2)$.
- (e) $\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{n(n+1)} = \frac{n}{n+1}$.
- (f) $\frac{1}{1 \cdot 3} + \frac{4}{3 \cdot 5} + \frac{9}{5 \cdot 7} + \dots + \frac{n^2}{(2n-1)(2n+1)} = \frac{n(n+1)}{2(2n+1)}$.

- (g) $\frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \frac{1}{3 \cdot 4 \cdot 5} + \cdots + \frac{1}{n(n+1)(n+2)} = \frac{n(n+3)}{4(n+1)(n+2)}.$
- (h) $11 | (2^{2n-1} + 3^{4n-2}).$
- (i) $30 | (7^{4n+1} - 7).$
- (j) $48 | (7^{2n} - 1).$

4. Nech $n_0 \in \mathbb{N}$. Dokážte, že pre každé $n \geq n_0$, $n \in \mathbb{N}$ platí:

- (a) $n! \geq 2^n$, $n_0 = 4$.
- (b) $2^n > n^2 + 2$, $n_0 = 5$.
- (c) $2^n > 2n$, $n_0 = 3$.
- (d) $3^n > n^3$, $n_0 \neq 3$.

1.3 Dirichletov princíp

Dirichletov princíp (niekedy tiež označovaný ako zásuvkový princíp, priehradkový princíp alebo princíp holubníkov) je matematické tvrdenie z teórie množín, prípadne kombinatoriky. Na rozdiel od typov dôkazov v predchádzajúcej kapitole nám tento princíp dáva iba odpoveď ÁNO/NIE. Nedáva konkrétne riešenie.

Veta 1.1 (Jednoduchá verzia) Nech $n, k \in \mathbb{N}$. Ak n je počet holubov a k je počet kliebok, pričom $k < n$, tak po rozmiestnení všetkých holubov do kliebok sú aspoň v jednej kliebke aspoň dva holuby.

Veta 1.2 (Všeobecná verzia) Nech $n, m \in \mathbb{N}$. Ak je viac ako mn holubov rozdelených do n kliebok, tak aspoň v jednej kliebke je viac ako m holubov.

Veta 1.3 (Zložitejšia verzia) Nech f je funkcia z konečnej množiny X do konečnej množiny Y ($f : X \rightarrow Y$). Nech $|X| = n$, $|Y| = m$ a $k = \lceil \frac{n}{m} \rceil$. Potom existuje aspoň k hodnôt $a_1, a_2, \dots, a_k \in X$ takých, že $f(a_1) = f(a_2) = \cdots = f(a_k)$.

Dôkaz. Nech $f : X \rightarrow Y$, $|X| = n$, $|Y| = m$ a $k = \lceil \frac{n}{m} \rceil$. Označme $Y = \{y_1, y_2, \dots, y_m\}$. Dôkaz vykonáme sporom. Predpokladajme, že tvrdenie vety nie je pravdivé. To znamená, že pre každé $i = 1, 2, 3, \dots, m$ existuje najviac ak $(k-1)$ prvkov $x \in X$ takých, že $f(x) = y_i$. To ale znamená, že definičný obor hodnôt funkcie f obsahuje najviac ak $m(k-1)$ hodnôt, teda $|X| \leq m(k-1)$. Z predpokladu $k = \lceil \frac{n}{m} \rceil$ vyplýva, že $k-1 < \frac{n}{m}$. Potom

$$m(k-1) < m \frac{n}{m} = n,$$

čo je v spore s tým, že $|X| = n$. Pravdivé je teda pôvodné tvrdenie a teda existuje aspoň k hodnôt $x_1, x_2, \dots, x_k$ takých, že

$$f(x_1) = f(x_2) = \dots = f(x_k).$$

□

Príklad 1.5 *Konferencie sa zúčastnilo 70 delegátov, ktorí hovoria 11 rôznymi jazykmi. Jedným jazykom hovorí najviac 15 delegátov. Organizačný výbor rozhodol, že za oficiálny bude považovať taký jazyk, ktorým hovorí najmenej 5 delegátov. Dokážte, že na konferencii boli aspoň 3 oficiálne jazyky.*

Kedže delegátov je 70 a hovoria 11 jazykmi, iste jedným jazykom hovorí nie menej, ako 5 delegátov. Teda existuje jeden oficiálny jazyk, nazvime ho jazyk A. Jazykom A hovorí najviac 15 delegátov, t.j., ostatnými 10 jazykmi hovorí aspoň 55 delegátov. Medzi týmito 10 jazykmi sa musí nájsť jazyk (označme ho B), ktorým hovorí najmenej 5 delegátov. To je druhý oficiálny jazyk. Jazykom B hovorí najviac 15 delegátov, teda ďalšími 9 jazykmi hovorí aspoň 40 delegátov. Zasa podľa Dirichletovho princípu možno spomedzi týchto 9 jazykov vybrať jeden oficiálny. Existenciu štyroch oficiálnych jazykov nie je možné dokázať.

1.3.1 Úlohy

1. V pivnici máme 4 fľaše červeného a 2 bieleho vína. Prišli hostia a priali si červené víno ochutnať. Bohužiaľ fľaše sú tmavé, bez označenia a v pivnici je tma. Koľko fliaš musíme vziať, aby medzi nimi bola aspoň jedna fľaša červeného vína? (A ak by sme chceli aspoň dve fľaše červeného vína?)
2. Do obchodu priviezli 25 debničiek jabĺk troch druhov (v debničke bol vždy len jeden druh). Dá sa tam určite nájsť 9 debničiek toho istého druhu?
3. Neporiadny študent má v zásuvke päť párov rôznych ponožiek. Koľko kusov ponožiek musí po tme vybrať, aby si bol istý, že medzi nimi budú dve rovnakej farby.
4. Na škole je 1000 žiakov a 30 tried. Dá sa dokázať, že aspoň v jednej triede bude aspoň 34 žiakov?
5. V triede je 33 žiakov. Jeden žiak urobil v diktáte 15 chýb. Nikto nemal viac. Majú aspoň traja žiaci taký istý počet chýb? A štyria žiaci?
6. V zásuvke je 10 červených, 8 zelených, 8 modrých a 4 žlté pastelky. Pastelky vyberáme so zatvorenými očami. Koľko ich musíme vziať, aby medzi nimi boli

- (a) aspoň štyri tej istej farby?
 - (b) aspoň jedna z každej farby?
 - (c) aspoň šesť modrých?
7. Za 23 dni máme prijímacie skúšky a chceme ešte prepočítať 93 úloh. Doplňte správne výroky: „Musíme v aspoň jeden deň vyriešiť aspoň . . . úloh.“
8. Koľko musí byť v skupine ľudí, aby sme mohli povedať, že aspoň dvaja
- (a) sa narodili v rovnakom dni v týždni.
 - (b) sa narodili v rovnakom mesiaci.
 - (c) majú narodeniny v rovnakom dni v roku?
9. Môžeme povedať, že na svete sú aspoň dvaja ľudia, čo sa narodili v tej istej minúte? (Predpokladajme, že nikto nie je starší ako 120 rokov.) A v tej istej sekunde?
10. Koľko musí byť v skupine ľudí, aby sme mohli určite tvrdiť, že v niektorom mesiaci v roku má aspoň päť z nich narodeniny?
11. Hádzeme dvoma kockami. Koľkokrát treba hodiť, aby sme mali zaručené, že dvakrát padol rovnaký súčet na kockách?
12. Koľkokrát treba hodiť tromi kockami, aby aspoň štyrikrát padol ten istý súčet.
13. Koľkokrát treba hodiť dvoma kockami, aby trikrát padla tá istá dvojica čísel.
14. Na konferencii sa zúčastnilo 40 delegátov, z 13 krajín. Dokážte, že delegácia aspoň jednej krajiny mala aspoň troch členov.
15. Konferencie sa zúčastnilo 70 delegátov, ktorí hovoria 11 rôznymi jazykmi. Jedným jazykom hovorí najviac 15 delegátov. Organizačný výbor rozhodol, že za oficiálny bude považovať ten jazyk, ktorým hovorí najmenej 5 delegátov. Dokážte, že na konferencii boli aspoň 3 oficiálne jazyky.
16. V pravidelnom dvadsaťuholníku je 9 vrcholov vyznačených zlatou farbou. Dokážte, že aspoň tri z nich tvoria rovnoramenný trojuholník.
17. V rovine je daných 17 bodov. Z nich žiadne tri neležia na tej istej priamke. Body sú spojené úsečkami žltej, červenej a modrej farby. Dokážte, že tak vznikne aspoň jeden „jednofarebný trojuholník“ (t.j. jeho strany sú úsečky rovnakej farby).

18. Máme štvorec so stranou dĺžky 10 metrov. V ňom je 101 bodov. Dokážte, že existuje trojuholník s obsahom $1 m^2$, v ktorom sú dva body.
19. V kocke s hranou 19 cm je 1332 bodov. Dokážte, že sú medzi nimi aspoň 2 body, ktoré majú vzdialenosť menšiu než 3.
20. Večer pred dážďom nám do chalupy nalietať veľa komárov. Spočítali sme tie, čo sa usadili na posteli s rozmermi $180\text{cm} \times 90\text{cm}$. Bolo ich 145. Bolo by aspoň teoretický možné, aby sme jednou ranou s plácačkou s rozmermi $15\text{cm} \times 15\text{cm}$ zasiahli aspoň tri z nich?
21. V priestore je 9 bodov s celočíselnými súradnicami. Dokážte, že vieme z nich vybrať aspoň 2 body tak, že stred nimi určenej úsečky má celočíselné súradnice.
22. Dokážte, že zo 100 prirodzených čísel vieme vždy vybrať niekoľko takých, že ich súčet je deliteľný 100.
23. Je daných $n + 2$ ľubovoľných kladných celých čísel. Dokážte, že medzi nimi existujú dve, ktorých rozdiel je deliteľný číslom $2n$, alebo dve, ktorých súčet je deliteľný číslom $2n$.
24. V rovine vyberáme niekoľko bodov s celočíselnými súradnicami. Spojíme ich úsečkami a určíme stredy týchto úsečiek. Koľko musíme vziať bodov, aby aspoň jeden stred úsečky mal celočíselné súradnice?
25. Dokážte, že z 50 ľubovoľne zvolených navzájom rôznych prvočísel vieme vždy vybrať 13 prvočísel tak, že rozdiel každých dvoch je deliteľný piatimi.
26. Dokážte, že ku každému prirodzenému číslu n existujú prirodzené čísla $r \neq s$ také, že $3^r - 3^s$ je deliteľné číslom n .
27. Cestovná kancelária skúmala podmienky šiestich miest a zistila, že každé dve z nich majú medzi sebou buď autobusové alebo vlakové spojenie (ale nikdy nie obe). Môžeme z toho usúdiť, že aspoň tri z týchto miest sú vzájomne prepojené rovnakým druhom dopravy?
28. Dokážte, že v každej skupine šiestich ľudí sú buď aspoň traja ľudia, ktorí sa navzájom poznajú, alebo aspoň traja ľudia, ktorí sa navzájom nepoznajú.
29. Dokážte, že v každej skupine aspoň dvoch ľudí existujú aspoň dvaja ľudia, ktorí majú rovnaký počet známych v danej skupine.
30. Majme 20 navzájom rôzne prepojených procesorov. Ukážte, že aspoň dva procesory sú spojené s rovnakým počtom iných.

1.4 Algoritmy

Pod pojmom **algoritmus** rozumieme postupnosť jednoduchých a jednoznačných krokov, ktorá nás dovedie k žadanému riešeniu konkrétneho problému.

Žiada sa, aby algoritmus mal tieto vlastnosti:

- **elementárnosť** – algoritmus sa skladá z jednoduchých (elementárnych) krokov, ktoré sú pre vykonávateľa (počítač, človek) zrozumiteľné.
- **determinovanosť** (jednoznačnosť) – každý krok algoritmu musí byť jednoznačne a presne definovaný. V každej situácii musí byť úplne zrejmé, čo a ako sa má vykonať. Uvedenú vlastnosť budeme požadovať od deterministických algoritmov. Pri rovnakom vstupe vykoná deterministický algoritmus vždy rovnakú postupnosť krokov a poskytne rovnaký výsledok. Existujú však aj nedeterministické algoritmy.
- **konečnosť** – vykonávanie procesu opísaného algoritmom skončí vždy po konečnom počte krokov.
- **rezultatívnosť** – po skončení algoritmus poskytne výsledok zodpovedajúci zadanému problému.
- **hromadnosť** – má byť použiteľný na celú triedu (vstupných) prípadov úlohy svojho typu.
Pri návrhu algoritmu sa navyše snažíme o jeho
- **efektívnosť** – algoritmus má zabezpečiť riešenie problému, v čo najkratšom čase použitím čo najmenšieho počtu prostriedkov.

Algoritmus možno zapísať slovne, pomocou pseudokódu, vývojového diagramu alebo programovacieho jazyka. V tejto učebnici budeme používať najmä pseudo-kód. Zápis algoritmu pomocou pseudokódu má mať tieto vlastnosti:

- jednoznačnosť. Musí byť jasný a presný,
- používame štandardné kľúčové slová (WHILE, DO, FOR, ELSE, ...),
- bloková štruktúra (viditeľnosť z diaľky),
- 1 veta = 1 riadok,
- premenné pomenúvame tak, aby bol ich význam zrozumiteľný.

Pre ten istý problém možno spravidla navrhnúť viacero rôznych algoritmov. Aby sme si z viacerých správne fungujúcich algoritmov mohli vybrať ten najvhodnejší, potrebujeme mať nejaké kritérium, podľa ktorého by sme mohli algoritmy

porovnávať. Algoritmy je možné hodnotiť z rôznych hľadísk. Napríklad podľa toho, aké ľahké je pre nás takýto algoritmus napísať. Kritériá, ktoré sa obyčajne používajú pri hodnotení kvality algoritmov, sa nazývajú časová a pamäťová zložitosť (alebo tiež efektívnosť) algoritmov.

Pamäťová zložitosť algoritmu definujeme ako závislosť pamäťových nárokov algoritmu na veľkosti riešeného problému. Vyjadrujeme ju ako počet premenných, resp. pamäťových miest, ktoré algoritmus pri výpočte so zvolenými vstupnými údajmi potrebuje. Pamäťová zložitosť algoritmu A je teda funkciou, ktorá veľkosti vstupných údajov priradzuje potrebný počet pamäťových miest, ktoré bude algoritmus A pri výpočte s týmito údajmi využívať.

Časová zložitosť algoritmu môžeme definovať ako závislosť časových nárokov algoritmu na veľkosti konkrétneho riešeného problému alebo konkrétnych vstupných údajov. Neudávame ju však v žiadnych bežných časových jednotkách, pretože skutočná doba výpočtu programu realizujúceho navrhnutý algoritmus závisí tiež na množstve čisto technických parametrov, ako je napríklad rýchlosť procesora použitého počítača, zatiaľ čo my chceme charakterizovať rýchlosť práce algoritmu samého. Budeme preto časovú zložitosť vyjadrovať počtom elementárnych operácií.

Budeme hovoriť, že **algoritmus vyrieši danú úlohu U v čase T** , ak na jej vyriešenie potrebuje T elementárnych krokov. Za dĺžku úlohy budeme považovať množstvo vstupných dát príslušnej úlohy. S istým zjednodušením možno za množstvo dát považovať počet čísel, ktorými je úloha charakterizovaná (bez ohľadu na ich veľkosť). Tak napríklad graf možno zadať n vrcholmi a m hranami, preto dĺžka každej úlohy o grafe bude $(n + m)$. V grafových štruktúrach budeme obyčajne brať za dĺžku úlohy $(n + m)$ aj keď pôjde o ohodnotené grafové štruktúry.

Funkcia $T(n)$ vyjadruje hornú hranicu počtu krokov algoritmu pre najhorší prípad úlohy s dĺžkou n . Nie je teda príliš dôležité, aké má algoritmus časové nároky pre malé hodnoty veľkosti vstupných údajov, pretože malé údaje bude takmer vždy možné algoritmicke spracovať v rozumnom čase. Podstatné je, ako sa chová funkcia vyjadrujúca časovú zložitosť algoritmov pre rastúce hodnoty. Hovoríme preto niekedy aj o **asymptotickej časovej zložitosti**. Časová zložitosť algoritmu býva obyčajne rastúcou funkciou. Z dvoch algoritmov budeme teda za časovo výhodnejší a efektívnejší považovať ten, ktorého funkcia časovej zložitosti s rastúcou hodnotou rastie pomalšie.

Príklad 1.6 *Napíšte pseudokód pre algoritmus INSERT Sort a vypočítajte jeho časovú a pamäťovú zložitosť.*

Algoritmus Insertion Sort na usporiadanie prvkov poľa $A = (a_1, a_2, \dots, a_n)$ do neklesajúcej postupnosti - InsertionSort:

- **Krok 1.** Polož $i := 2$.

- **Krok 2.** Ak $i > n$, STOP. Pole A je usporiadané. Inak pokračuj Krokom 3.
- **Krok 3.** Polož $x := a_i$ a $j := i - 1$.
- **Krok 4.** Ak $j \geq 1$ a zároveň $a_j > x$, polož $a_{j+1} := a_j$, $j := j - 1$ a zopakuj Krok 4. Inak pokračuj Krokom 5.
- **Krok 5.** Polož $a_{j+1} := x$.
- **Krok 6.** Polož $i := i + 1$ a choď na Krok 2.

Algoritmus postupne spracúva prvky na pozíciách $2, 3, \dots, n$. Prvok na pozícii i porovnáva s prvkami, ktoré sa nachádzajú pred ním. Počet vykonaných porovnaní preto závisí od pôvodného usporiadania poľa.

- Najlepší prípad nastane, keď je vstupné pole už správne usporiadané. Aktuálny prvok porovnáme iba s jeho bezprostredným predchodcom a zistíme, že ho netreba presúvať. Pre každý z $n - 1$ spracúvaných prvkov vykonáme 1 porovnanie. Celkový počet porovnaní je teda $n - 1$. Časová zložitosť v najlepšom prípade je preto $T(n) = O(n)$.
- Najhorší prípad nastane, keď je vstupné pole usporiadané zostupne. Každý nový prvok je menší ako všetky prvky pred ním, a preto ho musíme presunúť až na začiatok usporiadanej časti. Pri spracovaní druhého prvku vykonáme 1 porovnanie, pri spracovaní tretieho prvku 2 porovnania a pri spracovaní posledného prvku $n - 1$ porovnaní. Celkový počet porovnaní je $1 + 2 + 3 + \dots + (n - 1)$. Použitím vzorca pre súčet prvých $n - 1$ prirodzených čísel dostaneme $1 + 2 + \dots + (n - 1) = \frac{n(n-1)}{2}$ a po roznásobení dostaneme $\frac{n(n-1)}{2} = \frac{n^2 - n}{2}$.

Pri určovaní asymptotickej zložitosti ponechávame člen s najvyššou mocninou a zanedbávame konštantný násobok. Časová zložitosť v najhoršom prípade je preto $T(n) = O(n^2)$.

Insertion Sort usporadúva prvky priamo vo vstupnom poli. Okrem poľa používa iba niekoľko pomocných premenných, napríklad premennú x pre aktuálny prvok a jeho pozíciu. Počet pomocných premenných nezávisí od veľkosti vstupu. Pomocná pamäťová zložitosť je preto $M(n) = O(1)$. Ak do pamäťovej zložitosti započítame aj vstupné pole s n prvkami, celková pamäťová zložitosť je $O(n)$.

Definícia 1.2 Nech $g(n), h(n)$ sú dve nezáporné funkcie definované na množine prirodzených čísel. Budeme písať $g(n) = O(h(n))$ a hovoriť, že funkcia $h(n)$ asymptoticky dominuje funkcii $g(n)$, ak existuje konštanta $k \in \mathbb{N}$ a prirodzené číslo $n_0 \in \mathbb{N}_0$ také, že $g(n) \leq k \cdot h(n)$ pre $\forall n \geq n_0$. Hovoríme, že algoritmus A má zložitosť $O(f(n))$, ak pre horný odhad $T(n)$ počtu krokov algoritmu A a úlohu dĺžky n platí $T(n) = O(f(n))$. Skrátene hovoríme o $O(f(n))$ algoritme. Špeciálne

ak $f(n) \leq n^k$ pre nejaké konštantné $k \in \mathbb{N}$ hovoríme, že A je **polynomiálny** algoritmus. Hovoríme, že problém má zložitosť nanejvýš $O(f(n))$, ak preň existuje $O(f(n))$ algoritmus.

Majme problém P_1 , ktorý chceme vyriešiť, a problém P_2 , ktorý už riešiť vieme. Vstupné dáta D_1 problému P_1 najprv transformujeme na vstupné dáta D_2 problému P_2 . Následne vyriešime problém P_2 a získame riešenie R_2 . Ak vieme riešenie R_2 previesť na správne riešenie R_1 pôvodného problému P_1 , hovoríme o vhodnej transformácii problémov.

Transformáciu môžeme schematicky znázorniť nasledovne:

$$D_1 \xrightarrow{\text{transformácia vstupu}} D_2 \xrightarrow{\text{riešenie problému } P_2} R_2 \xrightarrow{\text{transformácia riešenia}} R_1.$$

Ak transformáciu vstupu D_1 na vstup D_2 a následný prevod riešenia R_2 na riešenie R_1 dokážeme vykonať v polynomiálnom čase vzhľadom na veľkosť vstupu, hovoríme o **polynomiálnej transformácii** alebo o **polynomiálnej redukcii** problému P_1 na problém P_2 . Túto skutočnosť zapisujeme

$$P_1 \leq_p P_2.$$

Zápis $P_1 \leq_p P_2$ znamená, že problém P_1 nie je výpočtovo ťažší ako problém P_2 . Ak poznáme polynomiálny algoritmus na riešenie problému P_2 , potom pomocou uvedenej transformácie získame polynomiálny algoritmus aj pre problém P_1 .

Pri redukcii je dôležitý jej smer. Zápis $P_1 \leq_p P_2$ znamená, že inštalácie problému P_1 transformujeme na inštalácie problému P_2 , nie naopak.

Príklad 1.7 Uvažujme nasledujúce dva problémy:

- P_1 – OPAKUJÚCI SA PRVOK: Zistiť, či sa v danom poli nachádza hodnota, ktorá sa v ňom vyskytuje aspoň dvakrát.
- P_2 – USPORIADANIE: Usporiadať prvky daného poľa vzostupne.

Ukážeme, ako môžeme problém P_1 vyriešiť pomocou algoritmu pre problém P_2 . Nech vstupom problému P_1 je pole $A = \{a_1, a_2, \dots, a_n\}$. Pole A odovzdáme ako vstup algoritmu na usporiadanie. Výsledkom bude usporiadané pole

$B = \{b_1, b_2, \dots, b_n\}$, pre ktoré platí $b_1 \leq b_2 \leq \dots \leq b_n$.

Ak sa niektorá hodnota v pôvodnom poli vyskytovala aspoň dvakrát, potom sa jej výskyty budú v usporiadanom poli nachádzať vedľa seba. Stačí preto skontrolovať, či existuje index $i \in \{1, 2, \dots, n-1\}$, pre ktorý platí $b_i = b_{i+1}$.

Celý postup môžeme schematicky zapísať ako

$$A \xrightarrow{\text{usporiadanie}} B \xrightarrow{\text{kontrola susedných prvkov}} \begin{cases} \text{áno,} & \text{ak existuje } i \text{ také, že } b_i = b_{i+1}, \\ \text{nie,} & \text{inak.} \end{cases}$$

Vstupom problému zisťovania opakujúcich sa prvkov je pole A . Toto pole môžeme priamo použiť ako vstup problému usporiadania. Ak by sme potrebovali vytvoriť jeho kópiu, trvalo by to $O(n)$ času.

Riešením problému usporiadania je usporiadané pole B . Z neho získame riešenie pôvodného problému tak, že postupne skontrolujeme všetkých $n - 1$ dvojíc susedných prvkov. Tých je $n - 1$. Táto kontrola trvá $O(n)$ času.

Transformácia vstupu aj prevod získaného riešenia teda vyžadujú najviac lineárny čas. Keďže lineárny čas je polynomiálny, ide o polynomiálnu redukciu problému zisťovania opakujúcich sa prvkov na problém usporiadania.

V príklade 1.7 sme ukázali, že pomocou algoritmu na USPORIADANIE prvkov vieme vyriešiť problém zisťovania OPAKUJÚCICH sa hodnôt. Platí teda

$$\text{OPAKUJÚCI SA PRVOK} \leq_p \text{USPORIADANIE}.$$

Redukcia má určený smer. Uvedený zápis znamená, že pomocou riešenia problému USPORIADANIE vieme získať riešenie problému OPAKUJÚCI sa prvok. V opačnom smere však uvedený postup nefunguje. Odpoveď na otázku, či pole obsahuje opakujúci sa prvok, nám neposkytuje dostatok informácií na usporiadanie celého poľa. Z existencie redukcie $P_1 \leq_p P_2$ vo všeobecnosti nevyplýva existencia redukcie $P_2 \leq_p P_1$. Každý smer redukcie sa musí navrhnúť a dokázať samostatne. Predpokladajme, že $P_1 \leq_p P_2$. Ak existuje polynomiálny algoritmus na riešenie problému P_2 , potom existuje polynomiálny algoritmus aj na riešenie problému P_1 . Opačné tvrdenie vo všeobecnosti neplatí. Ak poznáme polynomiálny algoritmus pre problém P_1 , zo vzťahu $P_1 \leq_p P_2$ nemôžeme usudzovať, že existuje polynomiálny algoritmus aj pre problém P_2 . Ak platí redukcia v oboch smeroch, teda

$$P_1 \leq_p P_2 \quad \text{a} \quad P_2 \leq_p P_1,$$

problémy P_1 a P_2 nazývame **polynomiálne ekvivalentné**. Túto skutočnosť môžeme zapísať ako

$$P_1 \equiv_p P_2.$$

V takom prípade existencia polynomiálneho algoritmu pre jeden z problémov implikuje existenciu polynomiálneho algoritmu aj pre druhý problém. Treba však zdôrazniť, že nie každý polynomiálny algoritmus je prakticky použiteľný. Algoritmus s časovou zložitou napríklad $O(n^{10})$ je síce z teoretického hľadiska polynomiálny, ale už pre pomerne malé vstupy môže byť príliš pomalý. Pri praktickom hodnotení algoritmu preto nestačí rozlišovať iba medzi polynomiálnou a nepolynomiálnou zložitou. Dôležitý je aj stupeň polynómu, konštantné faktory a veľkosť spracovávaných vstupov.

Trieda NP, z anglického Nondeterministic Polynomialtime, je trieda rozhodovacích problémov, pri ktorých možno správnosť kladnej odpovede overiť v polynomiálnom čase pomocou vhodného certifikátu.

Definícia 1.3 *Problém P nazývame **NP-ťažký**, ak možno každý problém z triedy NP polynomiálne redukovať na problém P . Problém P nazývame **NP-ľahký**, ak možno problém P polynomiálne redukovať na niektorý problém z triedy NP. Problém P nazývame **NP-úplný**, ak je súčasne NP-ťažký, NP-ľahký a patrí do triedy NP.*

Platí:

- **NP-ťažký**: všetky problémy z NP sa redukujú na P . V probléme obchodného cestujúceho je daná množina miest a cena cesty medzi každou dvojicou miest. Úlohou je nájsť okružnú cestu, ktorá navštívi každé mesto práve raz, vráti sa do počiatočného mesta a má najmenšiu možnú celkovú cenu. Tento problém je NP-ťažký. Neoznačujeme ho však za NP-úplný, pretože NP-úplnosť sa štandardne definuje pre rozhodovacie problémy. Jeho rozhodovacia verzia sa pýta, či existuje okružná cesta s cenou najviac k . Táto rozhodovacia verzia patrí do triedy NP a zároveň je NP-ťažká, preto je **NP-úplná**.
- **NP-ľahký**: problém P sa redukuje na problém z NP. Problém existencie cesty s dĺžkou najviac k možno riešiť v polynomiálnom čase. Patrí preto do triedy P, a teda aj do triedy NP. Ak platí $P \neq NP$, tento problém nie je NP-ťažký.
- **NP-úplný**: problém P patrí do NP a je NP-ťažký. Rozhodovací problém Hamiltonovskej kružnice patrí do triedy NP a zároveň je NP-ťažký.

NP-úplné problémy tvoria podtriedu triedy NP. Každý NP-úplný problém patrí do triedy NP, ale nie každý problém z triedy NP musí byť NP-úplný. Všetky NP-úplné problémy sú navzájom polynomiálne ekvivalentné. Ak by sme našli polynomiálny algoritmus pre jediný NP-úplný problém, potom by platilo $P = NP$. Doteraz sa podarilo pre stovky úloh dokázať, že sú NP-úplné. Pre ďalšie sa podarilo dokázať, že sú NP-ťažké. Pre žiadnu NP-úplnú úlohu sa doteraz nepodarilo nájsť polynomiálny algoritmus. Nepodarilo sa však ani dokázať, že polynomiálny algoritmus pre tieto úlohy neexistuje. Nájdienie takého algoritmu pre jedinú NP-úplnú úlohu by znamenalo existenciu polynomiálneho algoritmu pre všetky úlohy z triedy NP, teda platnosť rovnosti $P = NP$. Všeobecne sa predpokladá, že $P \neq NP$, toto tvrdenie však zatiaľ nebolo dokázané. Predpokladaná výpočtová náročnosť niektorých matematických problémov sa využíva aj pri návrhu kryptografických systémov.

Vzhľadom na to, že mnohé optimalizačné úlohy sú NP-ťažké, vznikajú aproximačné algoritmy a heuristiky. V praxi nám totiž často stačí dobré riešenie, ktoré síce nemusí byť optimálne, ale je blízke optimálnemu. Aproximačné algoritmy poskytujú

určitú záruku kvality nájdeného riešenia, zatiaľ čo heuristiky takúto záruku spravidla neposkytujú. Niektoré NP-ťažké problémy možno riešiť v polynomiálnom čase, ak vhodným spôsobom obmedzíme ich vstupy. Vo všeobecnosti možno heuristiky rozdeliť na **konštrukčné** a **zlepšovacie**.

Konštrukčná heuristika postupne vytvára riešenie tak, že začne najjednoduchším objektom, napríklad prázdnu množinou vrcholov, a potom tento objekt rozširuje podľa pravidiel, ktoré dáva nádej, že výsledné riešenie bude blízke optimálnemu. Jednou z možností je rozšíriť objekt o prvok, v teórii grafov najčastejšie o vrchol alebo hranu, ktorý sa v danom okamihu javí ako najvýhodnejší z hľadiska cieľovej funkcie. Takýto postup nazývame **pažravou metódou** (angl. *greedy method*). Pažravá metóda nemusí byť vždy iba heuristikou, pretože pri niektorých problémoch vedie k optimálnemu riešeniu.

Zlepšovacie heuristiky vychádzajú z existujúceho prípustného riešenia a snažia sa z neho jednoduchými operáciami získať susedné riešenie s lepšou hodnotou cieľovej funkcie. Takto pokračujú dovtedy, kým nedospejú k riešeniu, ktoré nemá žiadne lepšie susedné riešenie. Takéto riešenie nazývame lokálnym optimom a uvedený postup **lokálnym prehľadávaním** (angl. *local search* alebo *neighborhood search*). Lokálne prehľadávanie možno spustiť z niekoľkých náhodne vytvorených počiatočných riešení a za výsledok vybrať najlepšie z nájdených lokálnych optimálnych riešení.

Algoritmy možno často ďalej upravovať s cieľom znížiť ich časovú alebo pamäťovú zložitosť prípadne zlepšiť ich praktickú efektívnosť. Medzi základné možnosti optimalizácie algoritmov patria:

- predspracovanie vstupných údajov,
- predpočítanie často používaných hodnôt,
- ukladanie už vypočítaných výsledkov do pamäte,
- výpočet nových hodnôt pomocou predchádzajúcich výsledkov,
- odstránenie opakovaných a nepotrebných výpočtov,
- použitie vhodnejších dátových štruktúr,
- nahradenie rekúzie iteráciou, ak je to výhodné.