

# Úvod do informatiky

přednáška devátá

Miroslav Kolařík

Zpracováno dle učebního textu prof. Bělohávk:  
Úvod do informatiky, KMI UPOL, Olomouc 2008

- 1 Kombinatorika: princip inkluze a exkluze
- 2 Počítání pravděpodobnosti
- 3 Pojem algoritmu (intuitivní chápání)

- 1 Kombinatorika: princip inkluze a exkluze
- 2 Počítání pravděpodobnosti
- 3 Pojem algoritmu (intuitivní chápání)

Princip inkluze a exkluze je často používaný kombinatorický princip, který udává počet prvků sjednocení několika množin pomocí počtu prvků průniku jednotlivých množin.

## Věta: princip inkluze a exkluze

Pro množiny  $A_1, \dots, A_n$  platí

$$|A_1 \cup A_2 \cup \dots \cup A_n| = \sum_{\emptyset \neq I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|+1} \left| \bigcap_{i \in I} A_i \right|.$$

## Příklad

Ve městě M. fungují 3 kluby. Tenisový klub má 20 členů, kriketový klub 15 členů a egyptologický klub je osmičlenný. Přitom z egyptologů jsou 2 hráči tenisu a 3 hráči kriketu, tenis a kriket zároveň provozuje 6 lidí, a jediná obzvláště agilní osoba je ve všech třech klubech. Kolik osob se celkem účastní klubového života v M.?

**Řešení:**  $|T \cup K \cup E| =$

$$\begin{aligned} &= |T| + |K| + |E| - |T \cap K| - |T \cap E| - |K \cap E| + |T \cap K \cap E| = \\ &= 20 + 15 + 8 - 6 - 2 - 3 + 1 = 33. \end{aligned}$$

Klubového života ve městě M. se účastní 33 osob.

- 1 Kombinatorika: princip inkluze a exkluze
- 2 Počítání pravděpodobnosti
- 3 Pojem algoritmu (intuitivní chápání)

Představme si, že se koná nějaký pokus, který skončí jedním z výsledků  $e_1, \dots, e_n$ . Výsledkům  $e_1, \dots, e_n$  říkáme **elementární jevy**. Předpokládáme, že každý z výsledků  $e_1, \dots, e_n$  má stejnou šanci, tj. elementární jevy jsou stejně pravděpodobné. **Jev** je každá podmnožina  $A \subseteq \{e_1, \dots, e_n\}$ . **Pravděpodobnost**  $P(A)$  **jevu**  $A$  je dána vztahem

$$P(A) = \frac{|A|}{|E|},$$

tedy je to počet všech výsledků příznivých jevu  $A$  ku počtu všech možných výsledků.

Pravděpodobnost může nabývat reálných hodnot od 0 do 1; přitom 0 je pravděpodobnost **nemožného jevu**, 1 je pravděpodobnost **jistého jevu**.

### Příklad

Jaká je pravděpodobnost, že při hodu hrací kostkou padne

- a) šestka,
- b) číslo větší než jedna,
- c) sudé číslo,
- d) číslo deset,
- e) číslo menší než deset?

**Řešení:** Zřejmé.

### Příklad

Hodíme dvěma hracími kostkami. Jaká je pravděpodobnost, že

- a) na obou kostkách padne šestka,
- b) na obou kostkách padne liché číslo,
- c) bude součet bodů na kostkách menší než šest?

**Řešení:** Zřejmé.



### Příklad

Jaká je pravděpodobnost, že při hodu hrací kostkou padne

- a) šestka,
- b) číslo větší než jedna,
- c) sudé číslo,
- d) číslo deset,
- e) číslo menší než deset?

**Řešení:** Zřejmé.

### Příklad

Hodíme dvěma hracími kostkami. Jaká je pravděpodobnost, že

- a) na obou kostkách padne šestka,
- b) na obou kostkách padne liché číslo,
- c) bude součet bodů na kostkách menší než šest?

**Řešení:** Zřejmé.

## Příklad

Hodíme třikrát hrací kostkou. Jaká je pravděpodobnost, že

- a) padne právě jednou šestka,
- b) padne alespoň jednou šestka?

**Řešení:** Jednoduché,

a)  $3 \cdot \frac{1}{6} \cdot \frac{5}{6} \cdot \frac{5}{6} = \frac{25}{72},$

b)  $1 - \left(\frac{5}{6}\right)^3 = \frac{91}{216}.$

## Příklad

Z přirozených čísel 1 až 50 vybereme náhodně jedno číslo.  
S jakou pravděpodobností bude vybrané číslo

- a) dělitelné šesti,
- b) dělitelné šesti a osmi?

**Řešení:** Zřejmé.

## Příklad

Hodíme třikrát hrací kostkou. Jaká je pravděpodobnost, že

- a) padne právě jednou šestka,
- b) padne alespoň jednou šestka?

**Řešení:** Jednoduché,

a)  $3 \cdot \frac{1}{6} \cdot \frac{5}{6} \cdot \frac{5}{6} = \frac{25}{72},$

b)  $1 - \left(\frac{5}{6}\right)^3 = \frac{91}{216}.$

## Příklad

Z přirozených čísel 1 až 50 vybereme náhodně jedno číslo. S jakou pravděpodobností bude vybrané číslo

- a) dělitelné šesti,
- b) dělitelné šesti a osmi?

**Řešení:** Zřejmé.

### Příklad

Krychli o délce hrany 3 cm obarvíme červenou barvou a potom ji rozřežeme na krychličky o délce hrany 1 cm. Všechny malé krychličky zamícháme a náhodně vybereme jednu z nich. Jaká je pravděpodobnost, že vybraná krychlička

- a) bude mít obarvené právě 3 strany,
- b) bude mít obarvené právě 2 strany,
- c) bude mít obarvenou právě 1 stranu,
- d) nebude mít obarvenou žádnou stranu?

**Řešení:** Jednoduché.

### Příklad

Máme k dispozici 36 dobrých výrobků a 4 zmetky. Vybereme náátkou 9 výrobků. Jaká je pravděpodobnost, že mezi vybranými výrobky budou právě 2 nebo 3 zmetky?

**Řešení:** Jednoduché.

### Příklad

Krychli o délce hrany 3 cm obarvíme červenou barvou a potom ji rozřežeme na krychličky o délce hrany 1 cm. Všechny malé krychličky zamícháme a náhodně vybereme jednu z nich. Jaká je pravděpodobnost, že vybraná krychlička

- a) bude mít obarvené právě 3 strany,
- b) bude mít obarvené právě 2 strany,
- c) bude mít obarvenou právě 1 stranu,
- d) nebude mít obarvenou žádnou stranu?

**Řešení:** Jednoduché.

### Příklad

Máme k dispozici 36 dobrých výrobků a 4 zmetky. Vybereme námtkou 9 výrobků. Jaká je pravděpodobnost, že mezi vybranými výrobky budou právě 2 nebo 3 zmetky?

**Řešení:** Jednoduché.

- 1 Kombinatorika: princip inkluze a exkluze
- 2 Počítání pravděpodobnosti
- 3 Pojem algoritmu (intuitivní chápání)**

**Algoritmus** je přesný návod či postup, kterým lze vyřešit daný typ úlohy. (Algoritmem rozumíme předpis pro řešení "nějakého" problému. Jako jistý druh algoritmu můžeme chápat i např. kuchyňský recept. Serióznějším příkladem by bylo uvést například předpis pro konstrukci trojúhelníka pomocí pravítka a kružítka ze tří daných prvků nebo třeba algoritmus pro získání součinu dvou čísel.)

Algoritmus obsahuje

- 1) hodnoty vstupních dat
- 2) předpis pro řešení
- 3) požadovaný výsledek, tj. výstupní data.

Pro zpřesnění pojmu algoritmus tedy dodejme: je to předpis, který se skládá z kroků a který zabezpečí, že na základě vstupních dat jsou poskytnuta požadovaná data výstupní.

## Základní vlastnosti algoritmů

- **konečnost** (finitnost)

každý algoritmus musí skončit v konečném počtu kroků (v praxi je navíc chtěno, aby požadovaný výsledek byl poskytnut v "rozumném" čase (ne za milion let))

- **jednoznačnost** (determinovanost)

každý krok algoritmu musí být jednoznačně a přesně definován (v každé situaci musí být naprosto zřejmé, co a jak se má provést, jak má provádění algoritmu pokračovat)

- **obecnost** (hromadnost)

algoritmus neřeší jen jeden konkrétní problém, ale celou třídu obdobných problémů (např. nejen jak spočítat  $2 \cdot 6$ , ale např. jak obecně spočítat součin dvou celých čísel).



K dalším vlastnostem algoritmů patří:

**rezultativnost** – algoritmus při zadání vstupních dat vždy vrátí nějaký výsledek (např. chybové hlášení);

**korektnost** – výsledek vydaný algoritmem musí být správný;

**opakovatelnost** – při použití stejných vstupních údajů musí algoritmus dospět vždy ke stejnému výsledku.

**Příklady algoritmů:** Erathostenovo síto, Eukleidův algoritmus, Dijkstrův algoritmus, dělení mnohočlenů, vyřešení kvadratické rovnice, ...

**Poznámka:** Algoritmus můžeme chápat jako "mlýnek na data". Nasypeme-li do něj správná data a zameleme, obdržíme požadovaný výsledek. (Uvědomme si, že kvalita mlýnku může být různá – časová náročnost, paměťová náročnost, viz. dále.)

- rekurzivní algoritmy – využívají (volají) sami sebe
- hladové algoritmy – k řešení se propracovávají po jednotlivých rozhodnutích, která jsou nevratná; např. Kruskalův algoritmus pro hledání minimální kostry grafu
- algoritmy typu rozděl a panuj – dělí problém na menší podproblémy až po triviální podproblémy (které lze vyřešit přímo), dílčí řešení pak vhodným způsobem sloučí
- pravděpodobnostní algoritmy – provádějí některá rozhodnutí náhodně či pseudonáhodně
- paralelní algoritmy – rozdělení úlohy (třeba) mezi více počítačů
- genetické algoritmy – pracují na základě napodobování evolučních procesů, postupným "pěstováním" nejlepších řešení pomocí mutací a křížení

- algoritmy dynamického programování – postupně řeší části problému od nejjednodušších po složitější s tím, že využívají výsledky již vyřešených jednodušších podproblémů
- heuristické algoritmy – nekladou si za cíl nalézt přesné řešení, ale pouze nějaké vhodné přiblížení; používají se v situacích, kdy dostupné zdroje (např. čas) nepostačují na využití exaktních algoritmů (nebo pokud nejsou žádné exaktní algoritmy vůbec známy)

**Poznámka:** Jeden algoritmus může patřit zároveň do více skupin; např. quicksort s rekurzí je současně rekurzivní a typu rozděl a panuj.

Později (až definujeme potřebné pojmy) si ukážeme dva příklady algoritmů z teorie grafů (diskrétní matematika) a to Dijkstrův algoritmus a Kruskalův (hladový) algoritmus.