Kapitola 1

Úvod do problematiky

Co si na úvod procvičíme?

- Ukážeme si, jaké otázky si lze nad algoritmy pokládat, jak je řešit a jaké souvislosti lze při této práci nahlédnout
- Zopakujeme si, jak zjistit počet permutací nebo kombinací nad danou množinou
- Zopakujeme si rovněž základy počtu pravděpodobnosti, jak vypočíst pravděpodobnost průniku a sjednocení pro závislé i nezávislé jevy
- Budeme se zabývat podmíněnou pravděpodobností a Bayesovou větou
- Ukážeme si, jak počet pravděpodobnosti souvisí s algoritmy a informatickým přístupem k řešení problémů
- Poznáme, že některé pravděpodobností problémy lze řešit s využitím lemy o džbánu

1.1 Motivace

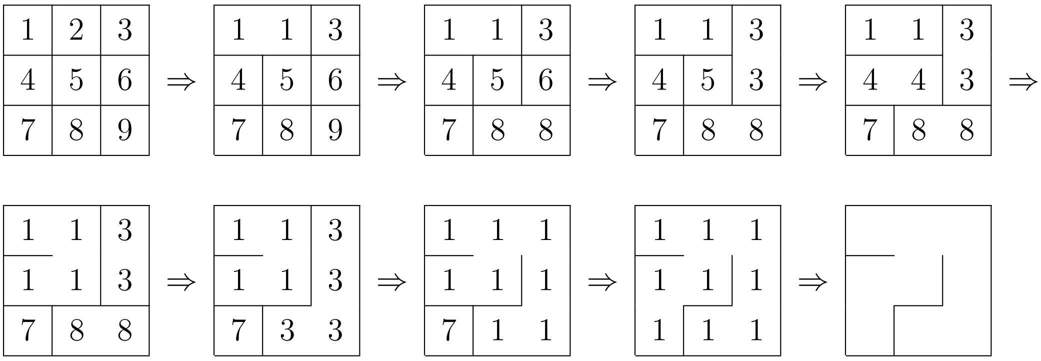
Ze začátku se podíváme na jednoduché úkoly, které jsou zamýšleny k řešení pro žáky základních škol. Na těchto příkladech chceme demonstrovat, jak lze nahlížet na problémy reálného světa a jak při řešení těchto problémů lze využít informatický přístup. Naznačíme, jaké typy úloh nás budou zajímat a jak k jejich řešení budeme přistupovat.

Úloha 1.1.1 (motivační[1]).
Algoritmus je popsán následovně:

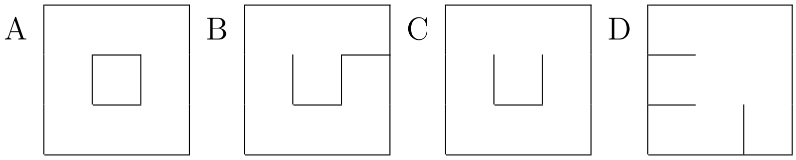
Krok A: Vytvoř čtvercovou mřížku. Každou buňku této mřížky nazýváme místnost. Všechny místnosti očísluj.
Krok B: Spoj dvě místnosti s různými čísly odstraněním jedné stěny.
Krok C: Očísluj novou (spojenou) místnost menším číslem z původních místností.
Krok D: Opakuj kroky B a C, dokud nezbude jedna místnost. Číslo této místnosti bude 1.
Krok E: Odstraň čísla.

Na obrázku je ukázka postupu podle tohoto algoritmu:

[1] Soutěž Bobřík informatiky. [online] Dostupné z: `https://www.ibobr.cz/o-soutezi/soutezni-otazky/11-soutezni-ulohy`



Který z následujících obrázků vznikl použitím tohoto algoritmu?



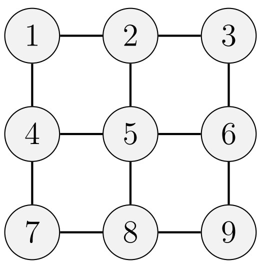
Odpovězte i na další otázky:

- Kolikrát se při tvorbě obrázku velikosti 3x3 zopakuje krok D algoritmu (odstranění stěny)?

- Kolik různých obrázků velikosti 3x3 lze podle daného algoritmu získat? Dokážete provést alespoň horní odhad?

  Řešení: Horní odhad je $\binom{12}{8} = \binom{12}{4} = 495$

- Definujme graf:

  Jednotlivé vrcholy grafu odpovídají jednotlivým buňkám mřížky. Předefinujte úlohu tvorby obrázku na odpovídající grafovou úlohu.

  Řešení: Problém souvisí s hledáním minimální kostry grafu.

- Vytvořte program pro hledání kostry grafu. Použijte metodu naznačenou v úloze – když spojíte dva vrcholy, označíte je stejným číslem, až postupně budou všechny vrcholy označeny jedničkou.

  Řešení: Algoritmus pro simulaci vyhledání jednoho řešení. Pokud použijeme postup naznačený motivační úlohou, dostáváme algoritmus UNION-FIND.

- Vytvořte program, který najde všechna řešení podle daného algoritmu (dané grafové úlohy). Kolik řešení jste dokázali vygenerovat? Porovnejte výsledek s vaším odhadem.

Řešení: Odhad 495, vygenerováno 192 různých možností.

- Řešte úlohu pro čtvercovou mřížku velikosti 2x2. Zamyslete se nad tím, jak se zvýšila náročnost algoritmu při změně velikosti vstupu – změna velikosti mřížky 2x2 na 3x3.

Řešení: Čtvercová mřížka velikosti 2x2 umožňuje 4 různá řešení úlohy.

- Odhadněte počet řešení úlohy pro mřížku velikosti 4x4. Svůj odhad si můžete opět ověřit programem. Jak dlouho bude výpočet probíhat?

Řešení: Horní odhad je $\binom{24}{15} = 1307504$.

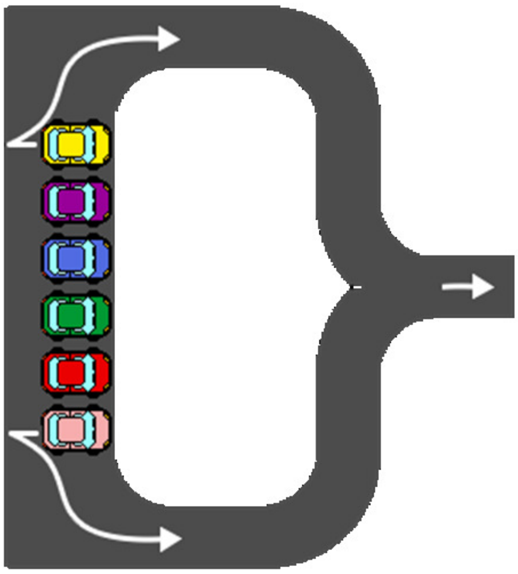
Skutečný počet koster nad daným grafem lze určit přes Laplaceovou matici (matice sousednosti, kde na diagonále je počet sousedů jednotlivých vrcholů a soused je značen hodnotou -1).

$$\begin{pmatrix}
2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2
\end{pmatrix}$$

Determinant Laplaceovy matice (určíme ho po odstranění jednoho řádku a jednoho sloupce - viz Kirchhoffova věta[2]) určí počet koster grafu. Počet koster uvažovaného grafu velikosti 4x4 je 100352.

- Srovnejte počet koster pro grafy - mřížky velikosti 2x2, 3x3, 4x4. Jak roste složitost úlohy s růstem velikosti mřížky?

Úloha 1.1.2 (motivační[3]).
Auta stojí v úzké garáži takto:



Z garáže může vycházet vždy jenom jedno z krajních aut. V jakém pořadí mohli auta z garáže vyjet? Vyberte jednu z možností na následujícím obrázku.

Tolik opět úloha pro žáky základní školy. Odpovězte na další otázky.

- Kolika způsoby mohou auta z garáže vyjet?

  Řešení: $2^6 = 64$ možností.

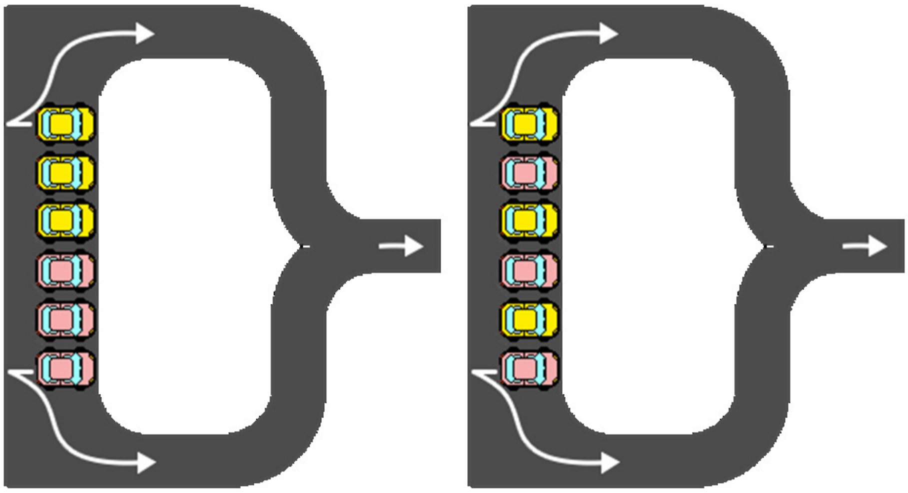
- Jsou všechny tyto možnosti unikátní, tj. neopakují se?

  Řešení: Poslední auto může vyjet horním nebo dolním výjezdem, na výsledném pořadí aut to nic nezmění. Ve skutečnosti je proto unikátních možností jenom $2^5 = 32$.

- Kolik je různých možností právě dvou modrých aut v garáži?

  Řešení: $\binom{6}{2} = \binom{6}{4} = 15.$

- Jak se změní počet možností, pokud některá auta obarvíme stejnými barvami? Řešte úlohu pro následující obrázky.



Úloha 1.1.3. Booleovské funkce

- Kolik existuje čísel, které mají dvojkový zápis s délkou n?

Řešení: 2^n.

- Kolik takových čísel existuje, pokud počítáme pouze čísla, která mají v nejvyšším bitu jedničku, tj. jejich dvojkový zápis délky n začíná jedničkou.

Řešení: 2^{n-1}.

- Dvojková čísla s délkou n jsou vstupem zadané funkce. Jejím výstupem jsou dvojková čísla s délkou m. Kolik takových funkcí existuje?

  Návod: Uvědomte si, že funkce definuje výstup pro každý ze svých vstupů. Definujte funkci, která má na vstupu dvojková čísla délky 2 a na výstupu dvojková čísla délky 1. Kolik takových funkcí existuje?.
  Řešení: Pro $n = 2$ a $m = 1$ existuje 16 různých funkcí.

1.2 Pravděpodobnost

Jevový prostor (označíme S) - množina všech elementárních jevů.
Náhodný jev (označíme A) je podmnožina jevového prostoru $A \subseteq S$.
Jistý jev - $A = S$.
Nemožný jev - $\Pr(A) = 0$
$\Pr(A) \geq 0$, $\Pr(S) = 1$, $\Pr(\emptyset) = 0$

$\Pr(A \cup B) = \Pr(A) + \Pr(B)$, je-li $A \cap B = \emptyset$
$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$

[2] Kovář, P. Teorie grafů. Ostrava, 2022. Strana 86-91. Dostupné z: `https://homel.vsb.cz/~kov16/files/skriptum_teorie_grafu_rozsirene.pdf`
[3] iBobor - súťaž: Archív úloh [online]. 2009/2010. Benjamín. Dostupné z: `http://ibobor.sk/sutaz_demo/`

Podmíněná pravděpodobnost je pravděpodobnost jevu A, za předpokladu, že nastal jev B ($\Pr(B) \neq 0$)

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

Nezávislé jevy - jevy, kdy skutečnost, že nastane jeden jev, nemá žádný vliv na to, zda nastane či nenastane jev druhý. Pravděpodobnost průniku nezávislých jevů A a B se rovná součinu pravděpodobností jednotlivých jevů, tj.

$$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$$

Bayesova věta

$$\Pr(A \cap B) = \Pr(B) \cdot \Pr(A \mid B) = \Pr(A) \cdot \Pr(B \mid A)$$

nebo jinak

$$\Pr(A \mid B) = \frac{\Pr(A) \cdot \Pr(B \mid A)}{\Pr(B)}$$

Úloha 1.2.1. Máme standardní hrací kostku. Vypočtěte následující pravděpodobnosti:

- $\Pr(\{3\})$
- $\Pr(\{2, 4, 6\})$
- $\Pr(\emptyset)$

$A = \{3, 4, 5\}, B = \{2, 3\}, C = \{1, 3, 5\}$

- $\Pr(A)$, $\Pr(B)$, $\Pr(C)$
- $\Pr(A \cap B)$
- $\Pr(A \cup B)$

Úloha 1.2.2. Jaká je pravděpodobnost, že při hodu dvěma kostkami je součet čísel 6? Jaká je pravděpodobnost, že součet je 7 nebo 9? Výpočet zkontrolujte pomocí programu, který bude situaci simulovat.

Úloha 1.2.3. V botníku je 12 párů bod. Jaká je pravděpodobnost, že vytáhneme právě tři boty na levou nohu?

Úloha 1.2.4. Výrobce uvádí, že jeho pevný disk dokáže fungovat 8000 hodin nepřetržitého provozu s pravděpodobností 95%.

- Určete pravděpodobnost nepřetržitého provozu 8000 hodin u diskového pole na serveru, které se skládá ze dvou takových pevných disků.
- Určete pravděpodobnost nepřetržitého provozu 8000 hodin u diskového pole na serveru, které se skládá ze čtyř takových pevných disků.

Úloha 1.2.5. Máme dvě permutace a a b čísel 1 až n. Jaká je pravděpodobnost, že následující funkce `f1`, `f2` a `f3` vrátí `True`?

```
def f1(n):
    return random.randint(0, n - 1) != 0
```

```
def f2():
    num = random.randint(0, 7) * 8 + random.randint(0, 7)
    return num == 45

def f3(a, b, n):
    ra = random.randint(0, n-1)
    rb = random.randint(0, n-1)
    return a[ra] == b[rb]
```

ÚLOHA 1.2.6. Na kostce padlo číslo menší než 4. Jaká je pravděpodobnost, že je liché?

ÚLOHA 1.2.7. Mince může být normální nebo falešná. Hodím dvakrát a pokaždé padne hlava. Jaká je pravděpodobnost, že mince je falešná? Jaká je pravděpodobnost, že je mince pravá?

1.3 Randomizované algoritmy

Randomizovaný algoritmus je algoritmus, který ve své logice využívá určitý stupeň náhodnosti. V praxi je "náhodnost"aproximována generátorem pseudonáhodných čísel, který je v počítači dostupný. Při analýze randomizovaných algoritmů je často potřeba využít počet pravděpodobnosti.

Lemma o džbánu Říká se, že se tak dlouho chodí se džbánem pro vodu, až se ucho utrhne. Následující lemma říká, že pokud označíme pravděpodobnosti toho, že se ucho utrhne, p, pak džbán vydrží $\frac{1}{p}$ pokusů. Formálně:

Lemma o džbánu: Čekáme-li na náhodný jev, který nastane s pravděpodobností p, dočkáme se ve střední hodnotě po $\frac{1}{p}$ pokusech.

ÚLOHA 1.3.1. Generujeme permutaci o n prvcích takto:

Krok A: Připravíme si n prázdných přihrádek.
Krok B: Postupně umisťujeme čísla $1, \ldots, n$ do přihrádek tak, že vždy vybereme náhodně přihrádku, a pokud v ní již něco je, vybíráme znovu.
Krok C: Do vybrané přihrádky umístíme číslo i.

Kolik pokusů budeme na vygenerování permutace v průměru potřebovat?

> *Návod:* Algoritmus implementujte a opakovaným spouštěním programu zjistěte průměrný počet pokusů pro $n = 10, 11, ..., 20$. Pokuste se o zobecnění výsledku.

ÚLOHA 1.3.2. Jaká je průměrná časová složitost bublinkového třídění? Přesněji: pokud dostaneme na vstupu náhodnou permutaci 1 až n, jaká bude střední hodnota doby výpočtu? Předpokládejme, že algoritmus se zastaví, jakmile během jednoho průchodu nic nepřehodí[4].

> *Návod:* Bublinkové řazení probíhá podle následujícího programu. Jaká je průměrná hodnota proměnné i těsně před skončením funkce `sort`? S využitím lemy o džbánu se pokuste o zobecnění výsledku.

[4]Fink, J. Algoritmy a datové struktury I, cvičení 11. Praha, 2019. Dostupné z: `https://ktiml.mff.cuni.cz/~fink/teaching/ads1/cviceni_11.pdf`

```python
def sort(arr):
    for i in range(len(arr)):
        is_change = False
        for j in range(len(arr)-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                is_change = True
        if not is_change:
            return
```

Úloha 1.3.3. Mějme počítač, jehož náhodným generátorem je ideální mince. Jinými slovy, máme instrukci `random_bit`, ze které na každé zavolání vypadne jeden rovnoměrně náhodný bit vygenerovaný nezávisle na předchozích bitech.
Jak pomocí takové funkce generovat rovnoměrně náhodná přirozená čísla od 1 do n? Minimalizujte průměrný počet hodů mincí (volání funkce `random_bit`)[5].

[5] Mareš, M. & Valla, T. Průvodce labyrintem algoritmů. Praha: CZ.NIC, Dostupné z: `http://pruvodce.ucw.cz/static/pruvodce.pdf`