

IV122 Zadání: Rozcvička

V průběhu hodiny se soustřeďte na úlohu A. Úlohu B je velmi vhodné mít zpracovanou nejpozději do příštího týdne. U úloh B, C, D jsou uvedeny „bonusy“ s otevřeným zadáním. Pokud některý z bonusů zpracujete, můžete jednu z úloh C, D vynechat.

A) Hrátky s čísly

Napište programy, které hledají odpovědi na následující otázky:

1. Které z přirozených čísel menších než 10000 má nejvíce dělitelů? Je odpověď na tuto otázku jednoznačná?
2. Některá čísla jdou vyjádřit jako součet tří druhých mocnin přirozených čísel, například $964 = 6^2 + 12^2 + 28^2$. Jiná takto vyjádřit nedou, například číslo 7. Kolik přirozených čísel menších než 1000 takto vyjádřit nejde?
3. Collatzova posloupnost je definována následovně: „vezmi přirozené číslo, pokud je sudé, vyděl jej dvěma, pokud je liché, vynásob jej třemi a přičti jedničku; tento postup opakuj, dokud nedostaneš číslo jedna“. Například pro číslo 27 potřebujeme 111 kroků, než se dostaneme na číslo 1. Pro které číslo menší než 10000 potřebujeme nejvíce kroků?
4. Jaký je součet všech prvočísel, která jsou menší než 1000 a neobsahují žádnou trojku?
5. Uvažme posloupnost, která začíná dvěma jedničkami a každý další člen je součtem dvou předchozích navýšený o jejich největšího společného dělitele. Posloupnost tedy začíná: 1, 1, 3, 5, 9, 15, 27, 45, 81, 135, 243, 405. Jaká je hodnota prvního prvku této posloupnosti, který je větší než milion?

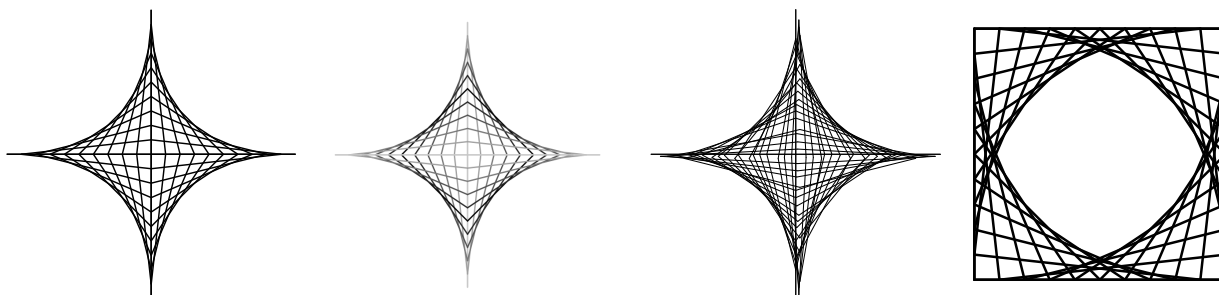
B) Práce s grafikou

Vytvořte si vlastní „knihovnu“ (případně najděte vhodnou existující) pro snadnou práci s vektorovou i bitmapovou grafikou.

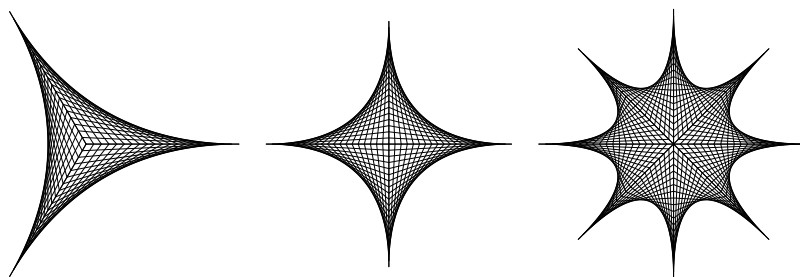
Práci s bitmapovou grafikou otestujte vytvořením následujícího obrázku:



Práci s vektorovou grafikou otestujte vytvořením hvězdy a několika jejích variací (ne nutně přesně těch uvedených). Vyvarujte se copy&paste kódu.



Bonus: Zobecněte program, aby vykresloval hvězdy s N rameny. Vymyslete další variace.



C) Ulamova spirála

Vytvořte program, který generuje Ulamovu spirálu. Vyzkoušejte, jak to dopadne, když místo prvočísel budeme do spirály zakreslovat čísla dělitelná k (vyzkoušejte např. 4, 5, 8). Soustřeďte se na elegantní zápis programu (vyvarujte se copy&paste kódu).

Bonus: Zkuste vymyslet vlastní zajímavou variaci na Ulamovu spirálu (jiné kritérium výběru polí, barevné obarvování, jiný než čtvercový tvar, ...).

D) Vizualizace NSD

1. Napište program generující obrázek vizualizující největší společné dělitele.
2. Napište program generující obrázek vizualizující délku běhu Euklidova algoritmu:
 - počet kroků algoritmu – odčítací varianta,
 - počet kroků algoritmu – efektivní modulo varianta,
 - různé způsoby barevného znázornění (např. kombinace obou předchozích do jednoho obrázku).

Bonus: Vytvořte vizualizaci pracující s rozkladem čísel na prvočísla.

IV122 Zadání: Kombinatorika, výpočty

A) Generování kombinací, permutací, variací

- Vstup: seznam, přirozené číslo k
- Výstup: výpis všech permutací, k prvkových kombinací (s opakováním), k prvkových variací (s opakováním)

Poznámky k implementaci:

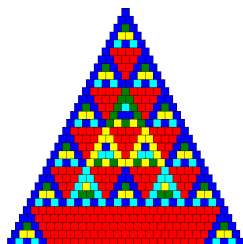
- nepoužívejte žádná připravená volání knihoven,
- je vhodné (a přirozené) použít rekurzi,
- jednotlivé varianty jsou myšlenkově podobné, zdrojový kód by to měl reflektovat.

B) Pascalův trojúhelník

Vstup: n, d

Výstup: Prvních n řádků Pascalova trojúhelníku, přičemž však nevypisujeme přímo čísla, ale znázorňujeme zbytek po dělení d .

Výstup zpracujte jako obrázek, zkuste najít vstupy, které dávají zajímavý výstup. Ukázka pro inspiraci (pro $n = 30, d = 5$):



C) Výpočet π

Implementujte několik metod pro aproximativní výpočet hodnoty π (minimálně Gregoryho-Leibnizova řada, Archimedova řada, Monte Carlo metoda, ideálně vyhledejte samostatně nějakou další). Experimentálně vyhodnoťte jednotlivé metody, např. jaké přesnosti jsou schopny dosáhnout během 1 vteřiny?

D) Umocňování

Vyberte si jednu z variant:

A) Přibližné umocňování: Pouze za použití základních aritmetických operací (+, -, *, /, porovnání) implementujte funkci pro přibližný výpočet x^y (pro kladná, ale ne nutně celá čísla x, y). Zkuste vymyslet a implementovat alespoň dvě rozdílné metody. Vyhodnoťte přesnost implementovaných metod – pro jaké hodnoty x, y dávají uspokojivé výsledky? Liší se jejich rychlost?

B) Efektivní modulo umocňování. Pouze za použití základních aritmetických operací implementujte výpočet $a^n \bmod k$ pro zadaná vstupní čísla a, n, k , přičemž n může být velmi velké (desítky cifer). Implementujte efektivní algoritmus s logaritmickou složitostí. Experimentálně porovnejte efektivní a naivní algoritmus.

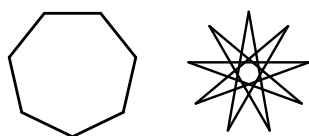
IV122 Zadání: Geometrie, želví grafika

Klíčovou součástí tohoto zadání je část D) Kreativita želvy. Tuto část určitě nevynechávejte.

A) Knihovna pro želví grafiku

Vytvořte vlastní „knihovnu“ pro práci s želví grafikou, tj. pro podporu minimálně následujících příkazů: `forward(step)`, `back(step)`, `right(angle)`, `left(angle)`, `penup()`, `pendown()`. Knihovna umožní také uložit výsledný obrázek do souboru (doporučeno použít SVG). Je doporučeno použít objektovou implementaci (třída `Turtle`).

Pro otestování knihovny vytvořte funkce pro vykreslení pravidelného mnohoúhelníku a různých „hvězdiček“.

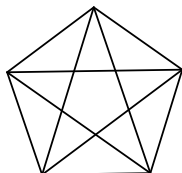


B) Vykreslování relativně vs. absolutně

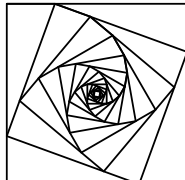
Želví grafiku můžeme chápat jako „relativní“ vykreslování – udáváme vždy, jak změnit polohu relativně k aktuálnímu bodu. „Absolutní“ vykreslování znamená, že udáváme absolutní souřadnice bodů, které se mají spojit (viz příklad ve slidech).

Vykreslete pentagram (obrázek A) oběma způsoby. U ostatních obrázků se zamyslete, který ze způsobů je pro vykreslení vhodnější, a ten použijte.

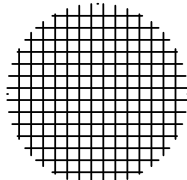
A



B



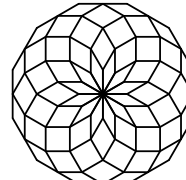
C



D



E

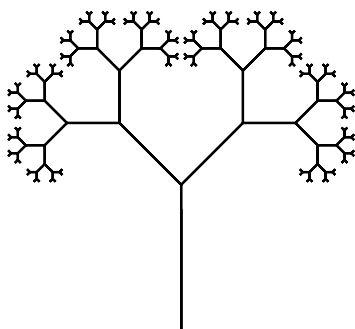


Zadání pokračuje na další straně.

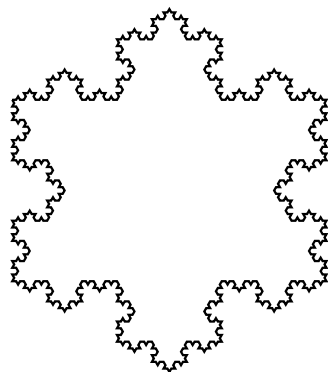
C) Želví grafika a fraktály

S využitím implementace A) vytvořte programy pro vykreslování fraktálů. Soustřeďte se primárně na Pětiúhelníkovou vločku, pro rozcvičení je ale vhodné začít prvním řádkem. Hilbertova křivka a Anklet jsou bonusové příklady. Vstupem programu je ve všech případech „hloubka zanoření“ obrazce.

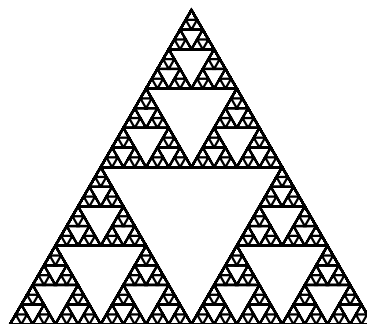
Keř



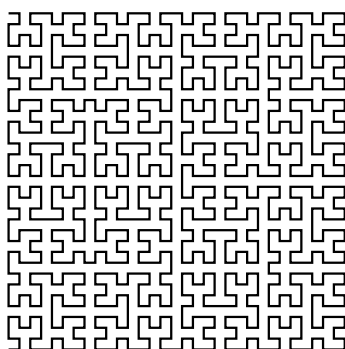
Kochova vločka



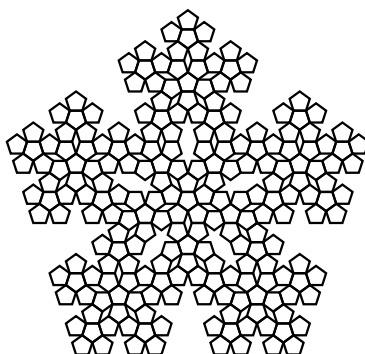
Sierpińského trojúhelník



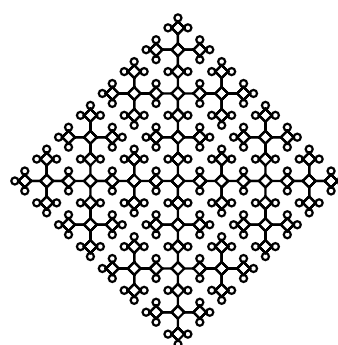
Hilbertova křivka



Pětiúhelníková vločka



Krishna Anklet



D) Kreativita želvy

S využitím želví grafiky vytvořte několik vlastních zajímavých obrázků, přičemž se snažte optimalizovat poměr „délka kódu / elegance obrázku“.

IV122 Zadání: Geometrie, bitmapová grafika

A) Základní útvary

Pomocí bitmapové grafiky (operace putpixel) vykreslete následující útvary (trojúhelník musí být rovnostranný).



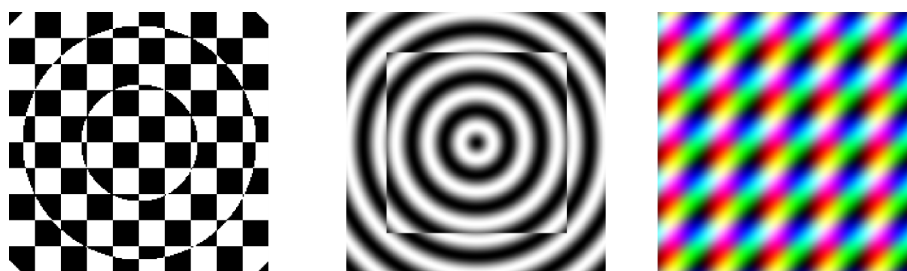
B) Mnohoúhelník

Napište program, který pomocí bitmapové grafiky (operace putpixel) vykreslí mnohoúhelník, který je zadáný jako seřazený seznam souřadnic vrcholů. Mnohoúhelník může být i nekonvexní. Příklad: (10, 10), (180, 20), (160, 150), (100, 50), (20, 180)



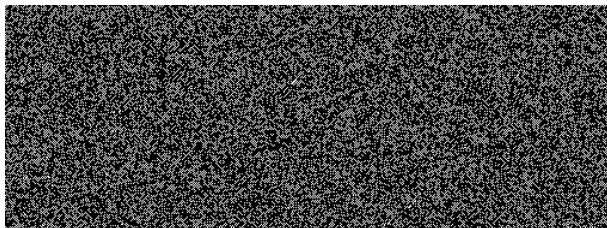
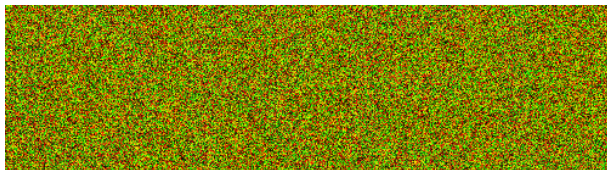
C) Efekty

Vygenerujte některé z následujících obrázků a vytvořte **vlastní zajímavé obrázky** podobného typu.



D) Bonus: Skrývačky

Následující obrázky skrývají zprávu, kterou dostanete vhodnou transformací vstupního obrázku (obrázky jsou dostupné samostatně ke stažení). Nápoředý: hledejte v modřé, hrany, xor s mřížkou. Jako tréning zkuste tyto skrývačky vyředit. Především ale vytvořte **vlastní zajímavou skrývačku**, pokud možno založenou na nějakém matematickém principu.



IV122 Zadání: Geometrické algoritmy

A) Průsečíky úseček

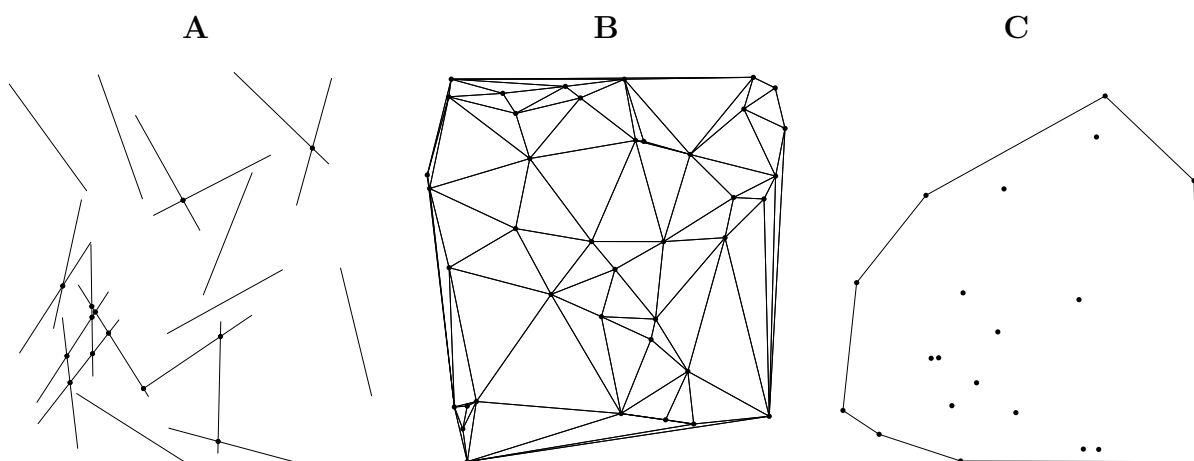
Napište program, který vygeneruje N náhodných úseček **stejné délky** a najde všechny jejich průsečíky.

B) Triangulace

Napište program, který vygeneruje N náhodných bodů v rovině a vykreslí jejich („hezkou“) triangulaci.

C) Konvexní obal

Napište program, který vygeneruje N náhodných bodů v rovině a vykreslí jejich konvexní obal.



Bonus (pro všechny tři úlohy): Použijte různé způsoby generování polohy bodů (úseček), například:

- náhodná uniformní distribuce,
- náhodná nerovnoměrná distribuce (např. normální rozložení podél osy x i y),
- pravidelná mřížka,
- polopravidelná mřížka (přidán náhodný šum, náhodně některé body vynechány).

Zkuste najít způsob generování polohy bodů, který vede k co nejzajímavějším výsledným obrázkům.

IV122 Zadání: Fraktály, chaos

A) Chaos game

Implementujte verzi „chaos game“ pro obecný n -úhelník a zadaný poměr r : máme n bodů v rovině, které tvoří pravidelný n -úhelník, začneme v náhodném bodě a v každém kroku jej posuneme směrem k jednomu náhodně vybranému vrcholu, délku posunu udává zadaný poměr r . Očekávané výstupy viz slidy.

Vyzkoušejte některé z následujících experimentů:

- Umístění bodů do jiné konfigurace než do pravidelného n -úhelníku.
- Přidělení různé „váhy“ bodům (rozdílné pravděpodobnosti při náhodném generování).
- Obarvení bodů na základě průběhu náhodnostního generování.

B) Feigenbaumův diagram

Napište program vykreslující Feigenbaumův diagram. Program by měl umožňovat výběr oblasti vykreslování (tj. „zoomování“ digramu), nemusí však být interaktivní.

C) L-systémy

Vytvořte program pro vykreslování fraktálů za použití L-systému a želví grafiky. Program by měl být generický, nikoliv specifický pro konkrétní L-systém, soustřeďte se na kompaktnost a eleganci implementace. Za použití tohoto přístupu vykreslete fraktály ze slidů (Kochova křivka, Sierpiňského fraktál, Hilbertova křivka, stromy) a alespoň 2 další fraktály (vlastní nebo inspirované z dostupných zdrojů). Proved'te experimenty s úhlem otáčení (např. otáčení o 91 stupňů místo 90).

Bonusy: 1) Přidejte podporu pro variabilní šířku a barvu čáry. 2) Implementujte stochastický L-systém.

Doporučený zdroj inspirace: *The Algorithmic Beauty of Plants*,
<http://algorithmicbotany.org/papers/abop/abop.pdf>

IV122 Zadání: Fraktály, komplexní čísla

Základ: Vytvořte programy pro vykreslování Newtonova fraktálu, Mandelbrotovy množiny a Juliových množin (princip viz slidy). Všechny programy by měly umožňovat vykreslovat zadaný „výřez“ množiny a měly by množiny vykreslovat barevně.

Tento základní úkol rozšířte alespoň v jednom z následujících směrů:

1. **Barvy:** Soustřed'te se na obarvení výsledného fraktálu – navrhňte a vyzkoušejte různé metody obarvení. Zdokumentujte použité postupy (popište, co jste zkoušeli, přiložte ukázky). Pokud se budete inspirovat z existujících zdrojů, uveďte citaci.
2. **Úprava funkcí:** Uvedené fraktály jsou založeny na vybrané funkci nad komplexními čísly ($z^3 - 1$, $z^2 + c$). Zkuste použít jiné funkce (např. jiné mocniny, přičtení konstanty). Vhodným způsobem „zmapujte“ výsledky.
3. **Rychlost:** Soustřed'te se na rychlost implementace – jak vykreslit co nej kvalitnější obrázek co nejrychleji. Zkuste využít paralelismus. Zdokumentujte použité postupy a uveďte statistiky rychlosti (především relativní porovnání jednotlivých postupů).
4. **Video:** Vygenerujte sérii obrázků a jejich spojením vytvořte video (animaci). Může jít třeba o „zoomování Mandelbrotovy“ nebo „animace Juliových množin pro plynule se měnící c “.

IV122 Zadání: Transformace v rovině

A) Afinity transformace

Implementujte „knihovnu“ pro afinity transformace v rovině a jejich aplikaci na rovinné obrazce. Transformace reprezentujte pomocí homogenních souřadnic (maticí 3×3). Zvolte vhodnou reprezentaci obrazce v rovině (např. seznam úseček) a implementujte především:

- generování základních transformací (např. funkce `rotation(angle)`, `scaling(sx, sy)`),
- skládání transformací,
- (opakovanou) aplikaci transformace na obrazec.

Snažte se o elegantní, modulární kód, který umožní snadné generování různých obrazců (bez zbytečného „copy&paste kódu“). Knihovnu otestujte na generování jednoduchých obrazců (viz příklady ve slidech).

B) Multiple Reduction Copy Machine (MRCM)

Implementujte MRCM (popis viz slidy) za použití implementace afinity transformací z úlohy A). Vstupem je počáteční obrazec, seznam transformací, počet iterací, výstupem je výsledný obrázek.

Za použití implementace vygenerujte následující obrázky:

- Sierpiňského trojúhelník,
- vybrané Sierpiňského příbuzné,
- kapradí (Barnsley fern, parametry podle slidů),
- hvězda (parametry podle slidů),
- alespoň 2 další zajímavé obrázky podle vlastního návrhu.

Pokuste se **vylepšit** vykreslování obrázků. Konkrétně: u hvězdy a kapradí zkuste dosáhnout hezčích obrázků oproti „přímočarému generování“, jehož výstupy jsou uvedeny ve slidech. Použitou metodu stručně popište.

IV122 Zadání: Pravděpodobnost

A) Monty Hall Problem

Popis problému viz slidy nebo například zde:

http://en.wikipedia.org/wiki/Monty_Hall_problem

Implementujte simulátor hry, vyzkoušejte strategie „zůstat při původním rozhodnutí“, „změnit rozhodnutí“, „náhodně měnit rozhodnutí“, experimentálně vyhodnoťte úspěšnost strategií v dlouhém běhu.

Alternativně můžete podobným způsobem zpracovat jinou podobnou úlohu (např. „girl/boy proportion problem“).

B) Nenáhodná čísla

Pro zadané posloupnosti čísel 1-6 určete, zda jsou „náhodné“ nebo „nenáhodné“. U „nenáhodných“ zdůvodněte, proč čísla nejsou náhodná (popište použité metody). Pokuste se odhadnout způsob, jakým byla vygenerována.

C) Centrální limitní věta

Uvažme následující dvě kostky (s čísly 1 až 6): K_a = zatížená kostka, která preferuje vyšší čísla (pravděpodobnost úměrná počtu teček), K_b = inverzně zatížená kostka. Pomocí kostek provedeme n hodů a vypočítáme z nich průměrnou hodnotu. Uvažme následující způsoby provádění hodů:

1. všech n hodů provádíme kostkou K_a ,
2. pro každý hod náhodně vybereme jednu z kostek K_a, K_b ,
3. náhodně vybereme jednu z kostek K_a, K_b a tou provedeme všech n hodů.

Celý tento proces opakujeme k krát. Jak vypadá distribuce průměrů pro jednotlivé možnosti? Jaký má tvar? Jde o (přibližně) normální distribuci? Jak tvar distribuce závisí na n a k ? Rozmyslete matematicky a vyzkoušejte experimentálně pomocí simulace.

D) Bayesova věta a simulace

V krabici je N kostek. Jedna z nich je falešná a má na všech stranách číslo 6, ostatní kostky jsou normální. Vytáhl jsem náhodně jednu z kostek, X krát jsem s ní hodil a pokaždé mi padla šestka. Jaká je pravděpodobnost, že jde o poctivou kostku? Úlohu řešte dvěma přístupy:

- Výpočtem pomocí Bayesovy věty.
- Simulací: opakovaně vytahujeme náhodnou kostku, házíme s ní a díváme se, jestli padla ve všech X hodech šestka.

Před tím, než začnete úlohu řešit, запиšte si svoje intuitivní odhady výsledku pro některé kombinace N a X (např. $N = 10, X = 5$; $N = 100, X = 3$; $N = 1000, X = 5$). Následně pro tyto hodnoty vypočítejte správná řešení a porovnejte s odhady.

IV122 Zadání: Analýza dat

Jako vstupy pro obě úlohy použijte poskytnutá data i vlastní simulovaná data, tj. dílčí podúkol je napsat generátor simulovaných dat.

A) Lineární regrese

Vstup: seznam bodů x_i, y_i .

Výstup: přímka $ax + b$ minimalizující sumu čtverců chyb; znázorněno obrázkem (body i přímka).

Algoritmy: výpočet vzorcem, aproximativní řešení (grid search nebo gradient descent).

U simulovaných dat prozkoumejte vztah mezi pravými hodnotami a, b (těmi, které byly použity pro generování) a vypočtenými hodnotami. Jak tento vztah závisí na množství dat a na velikosti šumu? Jak robustní je výpočet při přítomnosti odlehlých bodů (body s výrazně vyšší odchylkou než ostatní body)?

B) Detekce shluků

Vstup: seznam bodů x_i, y_i ; žádaný počet shluků k .

Výstup: rozdělení na k shluků; znázorněno obrázkem (obarvení bodů podle shluků).

Algoritmus: k-means.

Prozkoumejte chování algoritmu (stabilita, závislost na iniciálních podmínkách, počet iterací potřebných pro konvergenci) pro různé typy vstupních dat, konkrétně například pro různé počty shluků a míra „překrývání“ shluků.

IV122 Zadání: Grafy a bludiště

A) Číselné bludiště

Napište program pro řešení úlohy Číselné bludiště (pravidla a ukázka viz slidy). Program najde nejkratší řešení a **zkontroluje, zda je jednoznačné**, tj. zda existuje právě jedno řešení nejkratší délky. Výstup stačí textově (např. seznam souřadnic).

B) Variace na bludiště dle vlastní volby

Vyberte si jednu z „bludišťových variant“ a napište program pro řešení úlohy. Můžete si zvolit i jinou úlohu než ty, které jsou zmíněny na slidech (vlastní pravidla, jiná existující úloha). Úloha by jen měla souviset s grafy/bludišti a měla by být dostatečně obtížná (srovnatelně s diskutovanými variantami).

Formát vstupu si zvolte vlastní. Výstupem programu by mělo být snadno interpretovatelné „obrázkové“ řešení (minimálně v „textové grafice“, ideálně přes SVG nebo animaci).

IV122 Zadání: Generování bludišť

Napište program pro generování bludišť. Vyberte si jednu z následujících variant:

- perfektní bludiště na jiné než čtvercové mřížce,
- bludiště typu „braid“ (žádné slepé konce), může být na čtvercové mřížce, ale nemělo by obsahovat „náměstí“ (plocha 2×2 bez zdí),
- složitější „bludišťová variace“ (viz úlohy z předchozí hodiny).

Vstup programu: velikost bludiště, případně další parametry ovlivňující podobu bludiště.

Výstup programu: obrázek bludiště (např. SVG).

Program musí být náhodnostní (při každém běhu generuje jiné bludiště).

Bonusová otázka: Co ovlivňuje obtížnost zvolené úlohy? Záleží jen na velikosti zadání a délce nejkratší cesty?