Úvod do algoritmizace a složitosti

Úloha: Kevin a Scrabble

Algoritmy a datové struktury

Materiál s metodickými poznámkami

9. února 2026

Motivace: Kevinův problém

Zadání úlohy:

- Kevin hraje hru podobnou Scrabble.
- Má k dispozici dlouhý řetězec písmen.
- Každé písmeno má bodovou hodnotu (např. A=1, X=10).
- **Cíl:** Najít souvislý podřetězec délky K s nejvyšším součtem bodů.

Příklad

Řetězec: `X B T B B B X`, $K = 4$
Hodnoty: $X = 6, B = 1, T = 3$.

Která čtveřice vyhraje?

Metodická poznámka: Formalizace

Pro budoucí učitele

V této fázi se žáky **neřešte kód**. Cílem je převést slovní zadání do formálního modelu.

Klíčové otázky pro žáky:

- Co je vstupem? (Posloupnost znaků délky N, číslo K, tabulka hodnot).
- Co je výstupem? (Maximální součet, případně pozice začátku).
- Jak byste to řešili tužkou na papíře?

1. Naivní řešení (Brute Force)

Myšlenka: *"Vyzkoušíme všechny možnosti."*

```
MaxSoučet = 0

// Vnější cyklus: posouvá začátek okna
Pro i od 0 do (N - K):
    AktuálníSoučet = 0

    // Vnitřní cyklus: sčítá prvky v okně
    Pro j od i do (i + K - 1):
        AktuálníSoučet = AktuálníSoučet + Hodnota(Znak[j])

    Pokud (AktuálníSoučet > MaxSoučet):
        MaxSoučet = AktuálníSoučet
```

Analýza složitosti (Proč to nestačí?)

Představme si velká data:

- Délka textu $N = 100\,000$.
- Délka hledaného slova $K = 50\,000$.

Počet operací:

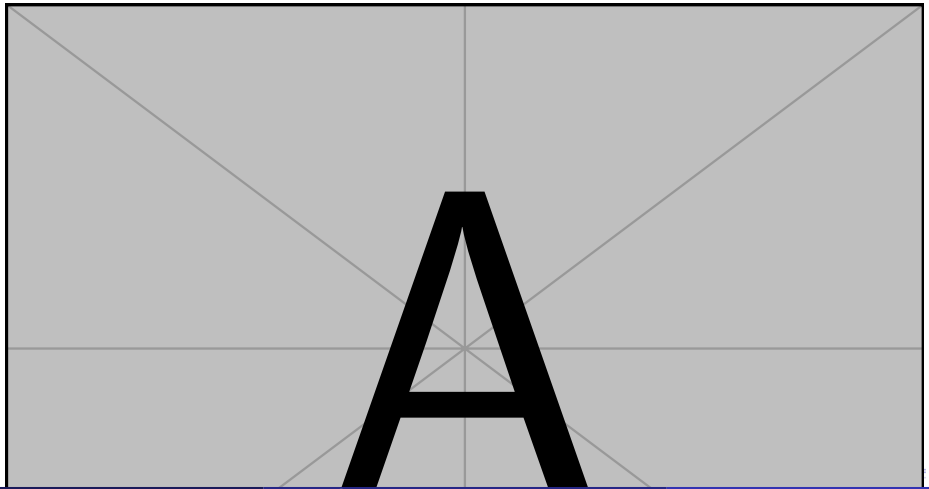
$$\text{Vnější cyklus} \times \text{Vnitřní cyklus} \approx (N - K) \times K$$

$$50\,000 \times 50\,000 = 2\,500\,000\,000 \text{ operací}$$

Závěr: Kvadratická složitost $O(N^2)$. Příliš pomalé!

Cesta k efektivitě: Sliding Window

Při posunu okna o jednu pozici doprava se většina obsahu nemění.



2. Efektivní řešení ($O(N)$)

```
// 1. Inicializace prvního okna
AktuálníSoučet = 0
Pro i od 0 do (K - 1):
    AktuálníSoučet = AktuálníSoučet + Hodnota(Znak[i])
MaxSoučet = AktuálníSoučet

// 2. Posouvání okna (Sliding Window)
Pro i od 1 do (N - K):
    // O(1) operace uvnitř cyklu!
    AktuálníSoučet = AktuálníSoučet - Hodnota(Znak[i-1])
                                    + Hodnota(Znak[i+K-1])

    Pokud (AktuálníSoučet > MaxSoučet):
        MaxSoučet = AktuálníSoučet
```

Srovnání přístupů

Vlastnost	**Naivní řešení**	**Sliding Window**
Počet cyklů	Dva vnořené	Dva za sebou
Složitost	$O(N \cdot K) \approx O(N^2)$	$O(N)$
Počet operací ($N = 10^5$)	$\approx 2,5$ miliardy	≈ 200 tisíc
Čas výpočtu	Sekundy až minuty	Okamžitě

Pedagogický cíl

Studenti musí pochopit, že **algoritmus** je důležitější než **hardware**. Rychlejší počítač naivní řešení nezachrání.

Závěrečné metodické pokyny

- **Gradace:** Nechte žáky nejprve vymyslet pomalé řešení. Ten moment "uvědomění si zbytečné práce"je klíčový.
- **Chyby:** Pozor na indexy (off-by-one errors) při odčítání prvku, který "vypadává"z okna.
- **Visualizace:** Použijte fyzické kartičky nebo magnetickou tabuli a posouvejte rámeček kolem písmen.