

Pole (Datová struktura)

Ladislav Vašina, xvasin11

xvasin11@stud.fit.vutbr.cz

17. dubna 2021

Pole (Array)

- Pole je nejjednodušší, nejstarší a nejčastěji používaná datová struktura v informatice.
- V paměti je uloženo jako lineárně uspořádaná posloupnost prvků stejného typu, ke kterým lze přistupovat pomocí tzv. **indexu**. [2]
- Ve většině programovacích jazyků velikost pole zůstává při běhu programu neměnná. [1]

Proč používat pole?

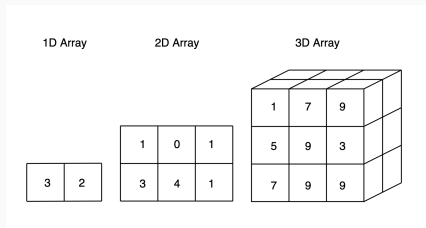
- Pole nám umožňují sdružovat daný vždy konečný počet prvků stejného datového typu.
- Pole mohou být použita pro řazení jejího obsahu.
- Pole se využívají při maticových výpočtech.
- Pole se využívají k implementaci dalších datových struktur (Stack, Fronta, Heap, atd.).

■ Jednorozměrná pole

- ☐ Každý prvek určen právě jedním indexem.
- ☐ Jednorozměrná pole se nejčastěji používají pro ukládání vektorů (pole čísel) a řetězců (pole znaků).

■ Vícerozměrná pole

- ☐ Matice a tabulky jsou příkladem polí dvojrozměrných.
- ☐ Každý prvek určují dvě souřadnice – index řádku a sloupce.
- ☐ Trojrozměrné pole je zase podobné Rubikově kostce.



Algoritmus 1: Inicializace jednorozměrného pole v Javě

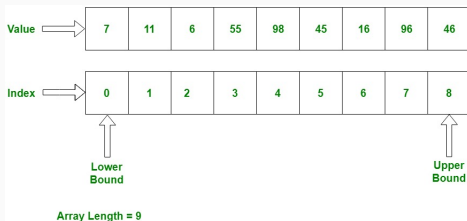
1: **int** pole1 [] = {1, 0, 5, -4, 55};

Algoritmus 2: Inicializace dvojrozměrného pole v Javě

1: **int** matice1 [][] = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};

Indexování polí

- Pomocí indexů přistupujeme k daným prvkům pole.
- **C, C++, C#, Java** a další indexují od nuly
- **BASIC** indexuje od jedničky, což odpovídá matematickému značení a přirozenému počítání
- **Visual Basic a Pascal** umožňuje nastavit horní a dolní meze pole individuálně



Ukázka indexování pole od 0

- **přístup k prvku** — probíhá v konstantním čase, tj. $\mathcal{O}(1)$; pomocí indexu lze vypočítat přesnou adresu v paměti
- **vyhledávání prvku** — (lineární vyhledávání) je řešitelné v lineárním čase, tj. $\mathcal{O}(n)$; v nejhorším případě je nutné projít celé pole
- **vyhledávání prvku v seřazeném poli** — hledání metodou půlení intervalu indexů pole (binární vyhledávání) má logaritmickou časovou složitost, tj. $\mathcal{O}(n \log n)$





■ Bubble sort

- ☐ Bubble sort je implementačně jednoduchý řadící algoritmus.
- ☐ Opakovaně prochází seznam, přičemž porovnává každé dva sousedící prvky, a pokud nejsou ve správném pořadí, prohodí je.
- ☐ Pro praktické účely je neefektivní. [3]

■ Merge sort

- ☐ Rozdělí neseřazenou množinu dat na dvě podmnožiny o přibližně stejné velikosti.
- ☐ Následně seřadí obě podmnožiny.
- ☐ Nakonec spojí seřazené podmnožiny do jedné seřazené množiny. [4]

Seznam použitých zdrojů

-  Pole (datová struktura). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit 2021-04-17]. Dostupné z: [https://cs.wikipedia.org/wiki/Pole_\(datová_struktura\)](https://cs.wikipedia.org/wiki/Pole_(datová_struktura))
-  Pole (array) [online]. [cit. 2021-04-17]. Dostupné z: <http://voho.eu/wiki/datova-struktura-pole/>
-  Bublinkové řazení. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-04-17]. Dostupné z: https://cs.wikipedia.org/wiki/Bublinkové_řazení
-  Řazení slučováním. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-04-17]. Dostupné z: https://cs.wikipedia.org/wiki/Řazení_slučováním