

# Data Structures and Algorithms (DSA)

## Dátové štruktúry a algoritmy

Lesson/Prednášky [lukas.kohutka@stuba.sk](mailto:lukas.kohutka@stuba.sk)

## Exercises/Cvičenie

[lukas.kohutka@stuba.sk](mailto:lukas.kohutka@stuba.sk)

[giang.nguyen@stuba.sk](mailto:giang.nguyen@stuba.sk)

[martin.komak@stuba.sk](mailto:martin.komak@stuba.sk)

[mirwais.ahmadzai@stuba.sk](mailto:mirwais.ahmadzai@stuba.sk)

[peter.lehoczky@stuba.sk](mailto:peter.lehoczky@stuba.sk)

[martin.sabo@stuba.sk](mailto:martin.sabo@stuba.sk)

[mohammad.momand@stuba.sk](mailto:mohammad.momand@stuba.sk)



<https://bit.ly/fiit-dsa-2023>

LS 2022/2023

# Data Structures and Algorithms (DSA)

Komunikácia/Communication **MS Teams (PM chat)**

<https://www.stuba.sk/navody/online/>

<https://elvira.fiit.stuba.sk>

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Po		-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,3)</sup> <i>M. Komák</i>	-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,3)</sup> <i>M. Komák</i>	-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,4)</sup> <i>M. Komák</i>	-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,4)</sup> <i>M. Komák</i>				-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(1,2)</sup> <i>M. Komák</i>	-2.01/a(CPUa) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,5)</sup> <i>M. Komák</i>		
		-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,7)</sup> <i>G. Nguyen Thu</i>	-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,7)</sup> <i>G. Nguyen Thu</i>	-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,8)</sup> <i>G. Nguyen Thu</i>	-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,8)</sup> <i>G. Nguyen Thu</i>				-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,6)</sup> <i>M. Sabo</i>	-2.01/b(CPUb) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,9)</sup> <i>M. Sabo</i>		
		-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,10)</sup> <i>M. Ahmadzai</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,10)</sup> <i>M. Ahmadzai</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,11)</sup> <i>M. Ahmadzai</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(2,11)</sup> <i>M. Ahmadzai</i>							
Ut												-1.61 (Aula Magna) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <i>L. Kohútka</i>
St			-2.01/d(CPUd) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(12)</sup> <i>G. Nguyen Thu</i>		-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <i>P. Lehoczky</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(13)</sup> <i>L. Kohútka</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(13)</sup> <i>L. Kohútka</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(14)</sup> <i>P. Lehoczky</i>	-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(14)</sup> <i>P. Lehoczky</i>		-2.01/c(CPUc) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(15)</sup> <i>M. Y. Momand</i>	
					-2.01/d(CPUd) (BA-FIIT-FIIT) Datové štruktúry a algoritmy <sup>(16)</sup> <i>M. Y. Momand</i>							

# Podmienky absolvovania

---

- Môžete získať celkovo 100 bodov
- Priebežne riešené zadania (max. 40 bodov):
  - na cvičení a doma sa budú riešiť 2 **zadania**
    - prvé za max. 20 bodov (min. 8)
    - druhé za max. 20 bodov (min. 8)
- Podmienky udelenia zápočtu:
  - žiadna neospravedlnená absencia z cvičení
  - celkovo minimálne 22 bodov zo zadaní
  - minimálny počet bodov za každé zadanie (8, 8)
- Záverečná skúška (max. 60 bodov)

Žiadna neospravedlnená absencia z cvičení – No absence in exercise  
Nie je povolené prevziať cudzí zdrojový kód a text – Plagiarism is not allowed

# DSA harmonogram – 12 weeks

- |     |   |   |
|-----|---|---|
| 1.  | <b><u>Prvé zadanie – 1st assignment</u></b>                                   |   |
| 2.  | Konzultácia/Consultation  |   |
| 3.  | Konzultácia/Consultation  |   |
| 4.  | Konzultácia/Consultation  |   |
| 5.  | <b>Code+PDF prezentácia/presentation</b><br><u>time slot 10 min / študent</u> |   |
| 6.  | <b>Code+PDF prezentácia/presentation</b><br><u>time slot 10 min / študent</u> | + |
|     | odovzdanie kódu a PDF do AISu   | + |
|     | deadline pred 6. cvikom   |   |
| 7.  | <b>Druhé zadanie – 2nd assignment</b>   |   |
| 8.  | Konzultácia/Consultation  |   |
| 9.  | Konzultácia/Consultation  |   |
| 10. | Konzultácia/Consultation  |   |
| 11. | <b>Code+PDF prezentácia/presentation</b><br><u>time slot 10 min / študent</u> |   |
| 12. | <b>Code+PDF prezentácia/presentation</b><br><u>time slot 10 min / študent</u> | + |
|     | odovzdanie kódu a PDF do AISu   | + |
|     | deadline pred 12. cvikom  |   |

# Zadanie/Assignment

1. week: Prvé zadanie – 1st Assignment

C/C++/C#/Java

7. week: **Druhé zadanie – 2nd Assignment**

–

Vizualization help for DSA students

# Plagiarism is **NOT** allowed

- Nie je povolené prevziať cudzí zdrojový kód!

Nie je povolené prevziať cudzí text!

Plagiarism is not allowed!

- Za copy/paste kódu a textu z hociakých zdrojov dostanete FX

Pokiaľ inšpirujete z verejných zdrojov a článkov →  
musíte tie zdroje citovať v kóde aj v správe o riešení zadania (PDF)

It is not allowed to copy someone else's source code. It is not allowed to copy someone else's text. Plagiarism is not allowed! For copy/paste code and text from any sources you will get FX. If you inspire from public sources and articles →you must cite those sources in both the code and the assignment solution report (PDF).

# Why DSA for Computer Science study?

8 DSA types: Array, Linked List, Stack, Queues, Searching, Sorting, Tree, Graph

## Efficiency

**DSA is the core foundation** to create a **good code** for real-world applications like (graphical) editors, simulators, games, libraries, drivers, data processing, ...

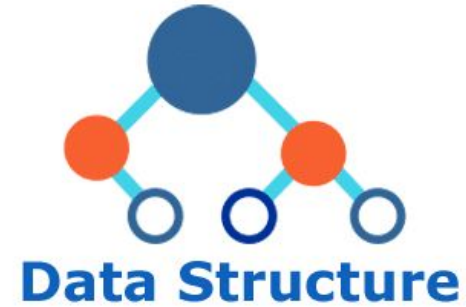
- Making your **code stable** as **attended** without bugs
- Making your **code fast** and **scalable**

Efficient usage of computing, memory resource and communication possibilities (time and space complexity)

# DSA is the baseline presumption for many areas

## Data Science and Artificial Intelligence

- **Thinking skill** – problem decomposition and algorithm design  
(the operations supported by data structures)
- Understanding the **complexity** of read/write/update times for operations  
(amount of computing resources)
- An **estimation** of the amount of memory that will be needed  
(amount of space resources)



## Accelerated computing (GPU/FPGA)

- Library, compiler and driver implementations

## Operating Systems and distributed computing

- Memory allocation
- Resource management: mapping, scheduling and load balancing
- Communication management





# Algorithms and programming languages

## Pseudo code (algorithm design) and Complexity

- Python
- Java
- C/C++/C#
- ~~JavaScript~~

**Integrated Development Environment (IDE)** – [JetBrains](#) – [Visual Studio Code](#)  
Free Professional version for FIIT STU students

- Python      [PyCharm](#)
- Java        [IntelliJ IDEA](#)
- C/C++/C#   [CLion](#)

# Learning source for DSA students

- **8 basic DSA types:** [PDF !!!](#)
- **Learn Data Structures and Algorithms**  
<https://www.programiz.com/dsa>
- **Data Structure Visualizations**  
<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
- **Algorithm Tutor**  
<https://algorithmtutor.com/>
- **GeeksforGeeks: Data Structures**  
<https://www.geeksforgeeks.org/data-structures/?ref=shm>

1., 2., 3., 4., 5. week  
Konzultácia – Consultation

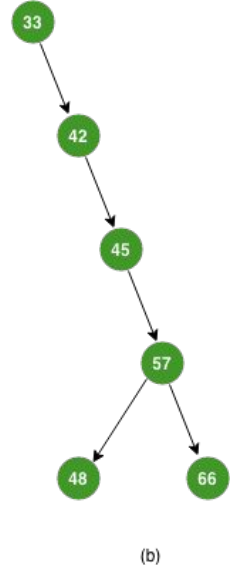
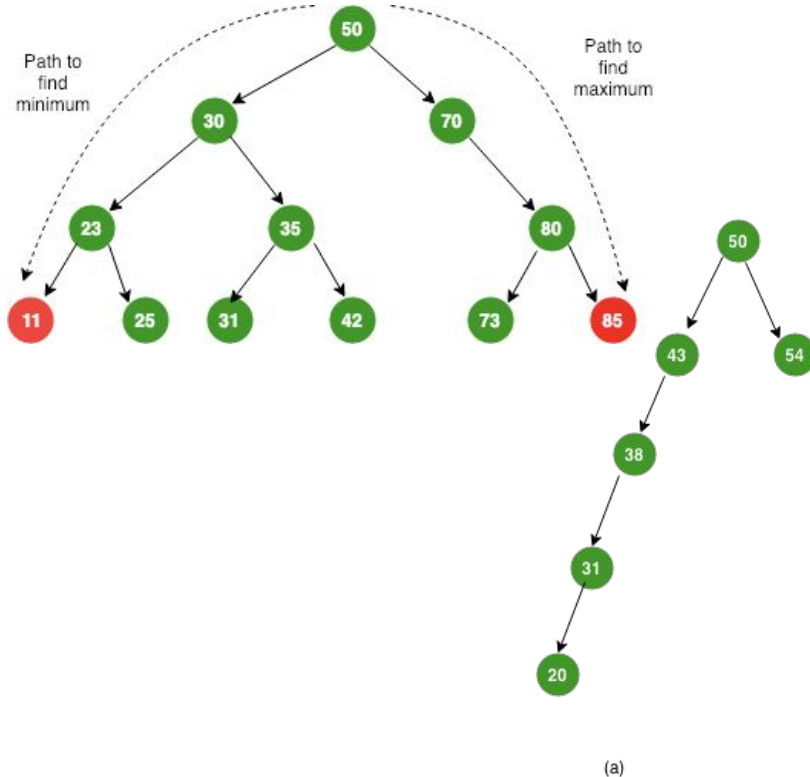
# BVS – Binárny vyhľadávací strom / Binary Search Trees – BST

## Basic operations

- Create
- **Insert**
- **Delete**
- **Search**

**Complexity problem of operations:** we want fast i.e. complexity  **$O(\log(n))$**  but tree(s) can be imbalanced  $\rightarrow$   **$O(n)$**

- AVL tree, splay tree, ...
- B tree,
- (a,b) trees: (2,3), (2,3,4)
- Red-Black trees, ...

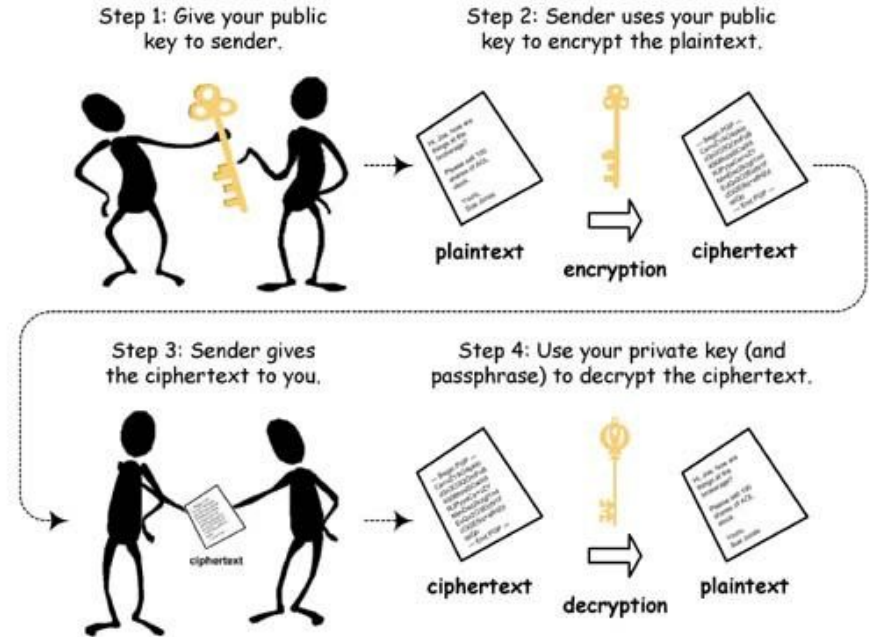


# Hashing – Encoding – Encryption

- **Encoding:** conversion of data from one format to another
- **Encryption:** conversion of an information into a cipher using keys, to maintain the confidentiality
- **Hashing:** ensuring the integrity of the data by converting it into a **fixed-length** string

## Popular Hashing Algorithms

- MD5 Message Digest (MD) Algorithm
- **SHA-256 Secure Hash Algorithm (SHA)**  
**SSH Login Without Password**
- CRC-32 Cyclic Redundancy Check (CRC)



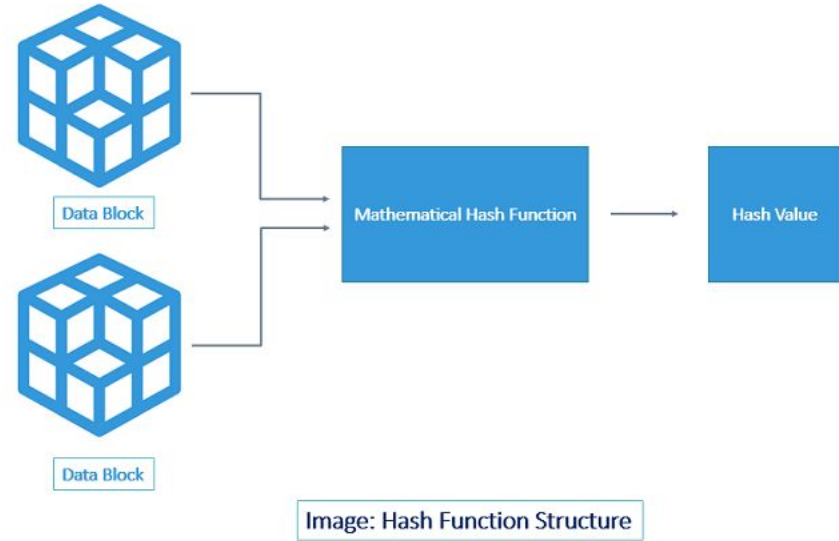
# Hash function

**Hashing** is defined by two distinct characteristics:

- **Irreversibility**: once you hash something, there is **no way back = one way encryption**
- **Uniqueness**: no two hash values are ever the same for two different pieces of data

## Hashing is used for text anonymization (GDPR)

In machine learning, **feature hashing**, also known as **the hashing trick** is a fast and **space-efficient way of vectorizing features**, i.e. turning arbitrary features into indices in a vector or matrix.



# Very nice explanations about BST/BVS + Hash

- [DSA\\_Stromy.pptx](#)
- [DSA\\_Hashovanie\\_family\\_friendly\\_edition.pptx](#)

Author: Bc. Adam Valach (DSA 2022)

Algorithms  
Pseudo code  
Data structures  
Time complexity – Space complexity



# Pseudo code vs. Algorithm

## Algorithm

- It's an organized logical sequence of the actions or the approach towards a particular problem.
- A programmer implements an algorithm to solve a problem.

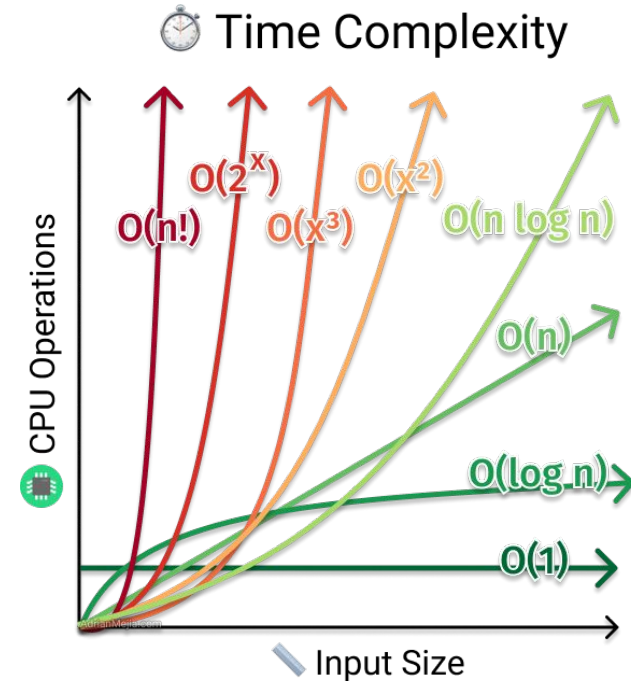
## Pseudo code

- It's simply an implementation of an algorithm in the form of annotations and informative text written in plain text.
- One of the best approaches to start implementation of an algorithm.

# Time complexity

**Big O** notation denotes the algorithms scalability and performance

- The worse case
- The best case
- The average case



# What does an algorithm use memory space for?

**Space complexity:** the memory amount needed for keeping data structure in a machine memory

- **Omega** Notation –  $\Omega$  describes an asymptotic lower bound
- **Theta** Notation –  $\Theta$  represents a function that lies within upper and lower bounds.



What does an algorithm use memory space for?



Notations used to express space complexity measurements



Big-O



Omega



Theta

What is space complexity?

## 5. week

# Konzultácia – Consultation

## Zber prvého zadania – 1st assignment submission

Odovzdanie zadanie v AIS sa uzavrie tesne pred 6. cvičením  
Assignment submission in AIS closes just before the 6th exercise

## 6. week

### Zber prvého zadania – 1st assignment submission

Odovzdanie zadanie v AISe sa uzavrie tesne pred 6. cvičením  
Assignment submission in AIS closes just before the 6th exercise

7., 8., 9., 10., 11. week  
Konzultácia – Consultation

# 11. week

## Konzultácia – Consultation

### Zber druhého zadania – 2nd assignment submission

Odovzdanie zadanie v AISe sa uzavrie tesne pred 12. cvičením  
Assignment submission in AIS closes just before the 12th exercise

## 12. week

### Zber druhého zadania – 2nd assignment submission

Odovzdanie zadanie v AIS sa uzavrie tesne pred 12. cvičením  
Assignment submission in AIS closes just before the 12th exercise



# Data Structures and Algorithms (DSA)

## Dátové štruktúry a algoritmy

Lesson/Prednášky [lukas.kohutka@stuba.sk](mailto:lukas.kohutka@stuba.sk)

## Exercises/Cvičenie

[lukas.kohutka@stuba.sk](mailto:lukas.kohutka@stuba.sk)

[giang.nguyen@stuba.sk](mailto:giang.nguyen@stuba.sk)

[martin.komak@stuba.sk](mailto:martin.komak@stuba.sk)

[mirwais.ahmadzai@stuba.sk](mailto:mirwais.ahmadzai@stuba.sk)

[peter.lehoczky@stuba.sk](mailto:peter.lehoczky@stuba.sk)

[martin.sabo@stuba.sk](mailto:martin.sabo@stuba.sk)

[mohammad.momand@stuba.sk](mailto:mohammad.momand@stuba.sk)



<https://bit.ly/fiit-dsa-2023>

<http://fiit.stuba.sk/~nguyen>