Algorithms and Data Structures II. Theoretical Questions

Ed. by **Tibor Ásványi**
Department of Computer Science
Eötvös Loránd University
Budapest, 2019.

asvanyi@inf.elte.hu

1. Data Compression. Character by character compression. Fixed-length code of characters. [1] 16.3

2. Data Compression. Prefix-free code. Huffman coding. The Huffman tree. Decoding the Huffman code. The optimality of the Huffman code. Is it the best? [1] 16.3; [4] 9.

3. Data Compression. Fixed-length code of character sequences. The Lempel–Ziv–Welch (LZW) algorithm. Encoding while building the dictionary. Decoding the LZW code while reconstructing the dictionary. [4] 9.

4. AVL trees, size and height, insertion, deletion, rebalancing (rotations). [2] 4.5; [5] 4.4

5. B+-trees, split keys, size and height, search, printing the data, insertion, deletion, splitting and merging nodes; B+-trees on disks. [3]

6. String Matching. The naive string-matching algorithm. Its run-time analysis. [1] 32.1

7. String Matching. The Quick Search algorithm. Its initialization. Its run-time analysis. [6]

8. String Matching. The Knuth-Morris-Pratt algorithm. Its initialization. Its run-time analysis. [1] 32.4

9. Graph representations. Breadth-first search (BFS). Shortest paths. Breadth-first trees. The run-time analysis of BFS. Printing a shortest path. [1] 22.1-2

10. Depth-first search (DFS). Depth-first trees, depth-first forest. Colors and timestamps of vertexes. Classification of edges, its connections with the colors and timestamps of the vertexes. The run-time analysis of DFS. [1] 22.3

11. Searching for directed cycles in graphs. The notion of DAG. Topological sort based on DFS. The asymptotic run-time of it. [1] 22.3-4

12. Minimum Spanning Trees (MSTs). Kruskal's algorithm. Its run-time analysis. [1] 23.1-2

11. Minimum Spanning Trees (MSTs). Prim's algorithm. Its run-time analysis. Representations of the priority queue. [1] 23.1-2

12. Single-Source Shortest Paths. Shortest-paths tree. Negative cycles [1]. The Queue-based Bellman-Ford algorithm (i.e. Breadth-first scanning). Its run-time analysis. [7] pp 92-93; [5] 9.3.3.

13. Single-Source Shortest Paths. Shortest-paths tree. The DAG shortest paths algorithm. Its run-time analysis. [1] 24.2

14. Single-Source Shortest Paths. Shortest-paths tree. Dijkstra's algorithm. Its run-time analysis. Representations of the priority queue. [1] 24.3

15. All-Pairs Shortest Paths. Matrix representation. Printing a shortest path. The Floyd-Warshall algorithm with the $D^{(k)}$ and $\Pi^{(k)}$ matrices. Its run-time analysis. [1] 25.2

16. Transitive closure of a directed graph. Transitive closure algorithm. Its run-time analysis. [1] 25.2

Notes:

At the exam the student must have only some pens, and his/her identity card with him/her. He/she may also have a non-smart watch, a bottle of liquid to drink, and some medicine, if needed, nothing else.

Each student receives two questions from the list above. They have enough time to take notes. Then they present their notes to the teacher. The teacher may pose questions about the student's notes.

Sources:

[1] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.,
. Introduction to Algorithms (Third Edition), *The MIT Press*, 2009.
[2] AD.pdf (in Canvas)
[3] B+trees.zip (in Canvas)
[4] Cormen, T.H.,
. Algorithms Unlocked, *The MIT Press*, 2013.
[5] Weiss, Mark Allen,
. Data Structures and Algorithm Analysis in C++ (Fourth Edition),
. *Pearson*, 2014.
[6] Quick Searching.ppt (in Canvas)
[7] Tarjan, Robert Endre,
. Data Structures and Network Algorithms, *Bell Laboratories*, 1987.