Algorithm Engineering -- EXERCISES
9 February 2024 – 1 hour

Name and Surname: #matricola:

Question #1 [score 4+3+3] Given the set of strings:

S = {AAAA, AACA, AACBB, AACBC, AACBDA, AACBDB, BAA, BAB, BBA, BBB},

index them via a **two-level indexing scheme** with block size of 2 strings, Front-coding compression in the blocks, and a Patricia trie in internal memory.

Then show how to perform:
- A **lexicographic** search for the position in S of the string: ABCBDB
- A **prefix** search in S for the string: AAC

Question #2 [score 3+3]. Given the sequence of integers (4, 5, 7), compress it via:
- Interpolative Coding
- Rice coding with k=2

Question #3 [rank 6]. Given the sequence (0 3 0 2 0 3 0) which is the result of the pipeline BWT+MTF+RLE with Wheeler code and RLE applied to runs of 1s (as in the class lectures), and given the MTF-list {A, B, R} and the $ position = 4, reconstruct the original string.

Question #4 [score 3+3]. Given the set of pairs:

{<3,5>, <9,3>, <13,9>, <5,1>, <10,4>, <4,8>}.

- Construct a Treap, using a MIN-heap, where the x-coordinate is the key, and the y-coordinate is the priority.
- Show the execution of a split at the key 6.

Question #5 [score 3] Apply the succinct encoding to the following binary tree: (a,b)(a,c)(c,d)(c,e), where node ‘b’ is to the left of node ‘c’, and node ‘d’ is to the left of node ‘e’. Then show how it works the navigation:

root → right child → left child

Algorithm Engineering -- THEORY
9 February 2024 – 45 minutes

Name and Surname: #matricola:

Question #1 [score 5+4]

- State the I/O-complexity of the multi-way mergesort in terms of M, B, and N
- Describe the disk-striping technique and state the I/O-complexity of the multi-way mergesort over D disks using Disk Striping

Question #2 [score 5+4+4].

- Describe why we introduced the Canonical Huffman
- Specify which data structures it keeps in the preamble of its compressed file
- Write its pseudo-code to decompress one symbol

Question #3 [rank 3 + 5]. Given two sorted lists of integers, say L1 and L2 of lengths n and m, respectively:

- Describe the “two-level memory” algorithm to compute their intersection
- State and prove its time complexity.