Advanced Algorithms for Data Science
Week 2 – Homework

Exercice 1. Red-Black trees

In the lecture, we studied the INSERT algorithm for red-black trees. In summary, INSERT starts by creating a new leaf X that is colored green ("problematic node"). The algorithm performs local rotations and recolorings which result either in the elimination of the green, or in the propagation of the green node upward. By iterating this step, we end up by completely getting rid of the green node.

Specific rotations and recolorings depend on the local configuration. The most interesting case occurs when X is not the root and the parent B of X is red (case 3 in the Lecture). Note that this implies that the parent of B exists and is black. This case further splits into two cases depending on whether the uncle of X is red (case 3.1 in the Lecture) or black (case 3.2).

Finally, case 3.2 splits into four subcases depending on whether B is the left or the right child of of its parent, and on whether X is the left or the right child of B. In the Lecture, we assumed that B is the left child of its parent and considered the two corresponding cases (3.2.1 and 3.2.2).

Question: *Describe the two symmetric cases arising when B is the right child of its parent.* In other words, describe the algorithm for the following cases

- the parent B of X is red, the uncle of X is black, B is the right child of its parent and X is the left child of B
- the parent B of X is red, the uncle of X is black, B is the right child of its parent and X is the right child of B

Exercice 2. Knuth-Morris-Pratt algorithm

A *period* of a string $T[1..n]$ is defined to be an integer p such that $T[i] = T[i+p]$ for all $1 \leq i \leq n-p$. For example, string *abaab* has period 3, string *abaaa* has period 4, string *abaaba* has periods 3 and 5, and string *abababa* has periods 2, 4 and 6. Obviously, each string has the minimal period.

1. Prove that p is a period of T iff $T[1..n-p] = T[p+1..n]$
2. Explain how the Knuth-Morris-Pratt algorithm can be used to compute the minimal period of a string.

Exercice 3. Suffix array

Recall that the occurrences of a pattern P in a text T correspond to an interval $[L_P, R_P]$ in the suffix array for T. In the lecture, we saw how to compute L_P by binary search. Write a binary search algorithm to compute R_P. (*Hint:* First, formulate accurately the definition of R_P and the condition that will guide the search.)