Problem 13.2

Musab Mehadi mmehadi@jacobs-university.de

Robin Karp string search algorithm basically uses hashes to determine specific matching strings from a given array.

Assuming we are using prime number P.

- We first calculate the hash value of our required substring
- X = old hash - value of old char
- X = X/P
- new hash = $X + P^{(n-1)}$ * new char, where n = size of substring
- If our hash matches we proceed to comparing the elements

The lines from the second step to the last are really helpful in that they help us to avoid computing the save values multiple times. (giving us a better time complexity)

Let's assume we are trying to find abc in abedabc

abc = $1*3^0 + 2*3^1 + 3(3^2)$ = 34

a = 1, b = 2, c = 3, d = 4, ..., z = 26 p = 3

abe = $1(3^0) + (2*3) + (5 \times 3^2)$ = 52 ≠ 34

bed = $\frac{52-1}{3} + (4*3^2)$ = 53 ≠ 34

eda = $\frac{53-5}{3} + (1*3^2)$ = 26 ≠ 34

dab = $\frac{26-5}{3}$ = 7 + 9*2 = 25 ≠ 34

abc = $\frac{25-4}{3} + (3*3^2)$ = 34 = 34 OK

Time complexity depends on the size of substring and the size of larger array O(nm).