

Problem 13.1

n_horse.cpp

Problem 13.2

a) Example of Rabin-Karp Algorithm

string text : 123456 text length N= 6 radix d = 10

string pattern: 456 pattern length M= 3

hash prime q: 13

transform "456" to one number: $4 \cdot d^{M-1} + 5 \cdot d^{M-2} + 6 \cdot d^{M-3} = 456$

hash value of "456" = $456 \% q = 1$

start from index 0:

hash value of "123" = $123 \% q = 6$ (doesn't equal to 1)

shift to next index 1: M-1

"234" = $(123 - 1 \cdot 10^{(3-1)}) \cdot 10 + 4$

this way efficiently combine digits into a number

hash value of "234" = $234 \% q = 0$ (doesn't equal to 1)

shift to next index 2:

hash value of "345" = $345 \% q = 0$ (doesn't equal to 1)

shift to next index 3:

hash value of "456" = $456 \% q = 1$ (equal to 1)

verify whether the 3-length substring starting from index 3 is what we want

the result is true, store index 3 into solution and keep the same process until

index is N-M, 3 in this case.

However, for strings with large radix d, we need to module after adding every character to the number. Here with radix 10, I did not use this method, but in (b) it will be shown.

b)

Rabin_Karp.cpp