Algorithms 2
Exercise 8 - sol

1. **Dictionary Matching** Consider the problem of finding the occurrences of any of a given set of $k$ patterns $P_1, \cdots, P_k$ in a text $T$. You may assume that the size of any pattern is $m$, and the size of $T$ is $n$.

   (a) Describe a method based on the KMP algorithm to solve this problem. What is the running time of this method?

   (b) Describe a method based on the Rabin-Karp algorithm to solve this problem. What is the (expected) running time of this method? Note that the algorithm should not return any false positives. Also, you may assume that the prime $q$ is at least $m$.

   (c) How would the previous answer change, if at all, given that the patterns all have different lengths?

**Answer** Sketch of the idea: For the kmp based algorithm, build a kmp automata for each pattern. Then run the normal kmp algorithm on each pattern. The time would be $O(nk)$. For the Rabin-Karp based algorithm, we can use hashing to reduce the time to $O(n)$ (instead of testing each pattern, we can test them all via one hash function).

2. **2D Matching** Consider the problem of finding the occurrences of a two-dimensional pattern $P$ of size $m \times m$, in a two dimensional text $T$ of size $n \times n$. Formally, we say $P$ occurs at location $(i, j)$ in $T$ if $\forall 1 \leq k \leq m, \forall 1 \leq \ell \leq m, p_{k, \ell} = t_{i+k-1, j+\ell-1}$.

   (a) Describe a method based on the KMP algorithm to solve this problem. What is the running time of this method?

   (b) Describe a method based on the Rabin-Karp algorithm to solve this problem. What is the (expected) running time of this method? (Previous notes apply, but may be adapted to this problem.)

**Answer** Sketch of the idea: For the kmp based algorithm, build a kmp automata for each pattern row. Then run the normal kmp algorithm on each row of the text, per each row in the pattern, and combine the possible answers. The time would be $O(n^2 m)$. For the Rabin-Karp based algorithm, we can use hashing to reduce the time to $O(n^2)$ (instead of testing each pattern row, we can test them all via one hash function).