CSCI3160: Special Exercise Set 7

Prepared by Yufei Tao

Problem 1. Given a string s of length n, stored in an array of characters, we call $s[i:j]$ a *substring* of s, for all pairs of i, j satisfying $1 \leq i \leq j \leq n$. Let x be a string of length n, and y a string of length m. Design an algorithm to find a longest common substring of x and y in $O(nm)$ time.

Problem 2*. Let $\boldsymbol{M}$ be an $n \times n$ matrix where each cell $\boldsymbol{M}[i,j]$ stores a distinct integer, for all $i \in [1,n]$ and $j \in [1,n]$. Define a *path* of length $\ell \geq 1$ to be a sequence of ℓ cells $\boldsymbol{M}[i_1, j_1]$, $\boldsymbol{M}[i_2, j_2]$, ..., $\boldsymbol{M}[i_\ell, j_\ell]$ satisfying both conditions below:

- for each $k \in [2,n]$, $\boldsymbol{M}[i_{k-1}, j_{k-1}]$ and $\boldsymbol{M}[i_k, j_k]$ are neighboring cells (this means the former cell is above, below, to the left of, or to the right of the latter cell);
- for each $k \in [2,n]$, $\boldsymbol{M}[i_{k-1}, j_{k-1}] < \boldsymbol{M}[i_k, j_k]$.

Design an algorithm that finds a path of the maximum length in $O(n^2 \log n)$ time.
(Hint 1: Find the length of longest paths starting from each cell)
(Hint 2: To choose a topological order, sort all the cells)

Problem 3.** Improve the running time of your solution to Problem 4 to $O(n^2)$.
(Hint: What is the dependency graph among the cells?)