EECS 340/454: Algorithms Spring 2020

ASSIGNMENT 4: SORTING IN LINEAR TIME

Instructors: Mehmet Koyutürk and Orhan Özgüner Due: March 2 before 11:59 PM

Problem 1

Draw the decision tree for the deterministic version of QUICKSORT on an array with $n = 3$ elements.

Problem 2

Given an array A of n integers in the range $[0, k)$, we would like to build an index, which we would like to use to answer any query of type "what is the number of integers in A that are in the range $[a, b]$?" For this purpose, write the following two procedures:

- Procedure PREPROCESS(A) should process A to build an index in $O(n + k)$ time. The size of your index should be $O(k)$.
- Procedure QUERY(A, a, b) should return the number of integers in A that are in the range $[a, b]$ in $O(1)$ time.

Problem 3

Let A be an $m \times n$ matrix. The *transpose* of A is an $n \times m$ matrix A' such that for all $0 \leq i < n$ and $0 \leq j < m$, $A'(i, j) = A(j, i)$. For example, if

$$A = \begin{bmatrix} 0 & 9 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 8 & 0 & 0 \end{bmatrix} \tag{1}$$

then

$$A' = \begin{bmatrix} 0 & 0 & 3 \\ 9 & 0 & 8 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{2}$$

Let k denote the number of non-zero entries in $m \times n$ matrix A (in the above example, $k = 5$). We say that a matrix A is *sparse* if $k = o(mn)$. Clearly, it is more efficient to store a sparse matrix using a special data structure, instead of storing it as a 2-dimensional array. One common data structure is known as the *compressed sparse-row* (CSR) representation.

In CSR representation, matrix A is stored using three arrays: R, C, and V. These arrays are respectively of length $m + 1$, k, and k. The array C stores the column indices of the non-zeros, such that for each $0 \leq i \leq m - 1$, the subarray $C[R[i]..R[i + 1] - 1]$ stores the column indices of the ith row of A, in increasing order (for convenience, the indexing of the arrays starts from 0). For each $C[j]$, $V[j]$ stores the value of the corresponding non-zero entry, i.e., if $R[i] \leq j < R[i+1]$, then $V[j] = A(i, C[j])$. For example, the CSR representation of matrix A in Equation (1) is as follows:

$$\begin{aligned} R = & \ < 0, 2, 3, 5 > \\ C = & \ < 1, 3, 2, 0, 1 > \\ V = & \ < 9, 1, 1, 3, 8 > \end{aligned} \tag{3}$$

For this problem, you are asked to write the algorithm for transposing a matrix in CSR representation. In other words, write the pseudo-code of procedure SPARSE-TRANSPOSE(R, C, V, m, n, k) that will return arrays R', C', and V', representing the transpose of the matrix represented by R, C, and V in row major format. For example, if your procedure is called with the R, C, and V in Equation (3), it should return:

$$\begin{aligned} R' = & \ < 0, 1, 3, 4, 5 > \\ C' = & \ < 2, 0, 2, 1, 0 > \\ V' = & \ < 3, 9, 8, 1, 1 > \end{aligned} \tag{4}$$

which is the row-major representation of A' in Equation (2). Your algorithm should work in $O(m + n + k)$ time. (*Hint:* The algorithm can be thought of as a generalization of COUNTING-SORT).