McGill University COMP251: Assignment 3

Worth 10%. Due October 29 at the beginning of lecture (10am sharp!)

Question 1 Give an algorithm that sorts (into non-decreasing order) an input array of n integers in the range 0 to $n^3 - 1$. Your algorithm must run in $\mathcal{O}(n)$ time.

Question 2 For this question, an arithmetic expression (or just expression) is built from integers and variables $x_1, x_2, \ldots$, using the operations $+, -, \times, \div$, and the parentheses (,) as follows:

- any number is an expression,
- any variable is an expression,
- if A and B are expressions, then so are $(A - B)$ and $(A \div B)$,
- if A_1, A_2, …, A_k are expression, then so are

$$(A_1 + A_2 + \ldots + A_n) \qquad \text{and} \qquad (A_1 \times A_2 \times \ldots \times A_n)$$

For example,

$$((x_1 + 5 + (x_2 \times 3 \times x_6) + (x_2 \div x_1)) - 4)$$

is an expression.

(a) Give a data structure for representing arithmetic expressions as trees of unbounded branching. Clearly explain the fields you are using.

(b) Give an algorithm that on input (A, X), where A is the root of the tree representing an expression which we also call A and X is the array of the values for variables, outputs the value of expression A when the variables are set according to X (i.e., $x_1 = X[1], x_2 = X[2]$, etc.).

(c) Give an algorithm that given the root of the tree representing an expression prints out the expression.

Question 3 There are two types of professional wrestlers: "babyfaces" ("good guys") and "heels" ("bad guys"). Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have n professional wrestlers and we have a list of r pairs of wrestlers for which there are rivalries. In this question, you are asked to give an $\mathcal{O}(n + r)$-time algorithm that determines whether it is possible to designate some of the wrestlers as babyfaces and the remainder as heels such that each rivalry is between a babyface and a heel. If it is possible to perform such a designation, your algorithm should *print* it.

The input to your algorithm is an array W of distinct names (of the wrestlers), and an array R of *distinct* pairs of rivalries. *The pairs in R are pairs of indices: if (i, j) is in R, then $W[i]$ and $W[j]$ are rivals.*

(a) Clearly describe the data structure you are using.

(b) Give the algorithm. (*Your algorithm should consist of 3 parts: one for parsing the input, one for performing some graph search, one for printing the output*).

(c) Verify that your algorithm runs in time $\mathcal{O}(|W| + |R|)$, where $|W|$ and $|R|$ denote respectively the lengths of the arrays W and R.

(d) Prove that your algorithm is correct.

Question 4 Let $G = (V, E)$ be a directed graph, in which each vertex $v \in V$ is labeled with a unique integer $L(v)$ called the label of v. *Each label $L(v)$ is an integer between 1 and $5|V|$.* For each vertex v, let $R(v)$ be the set of vertices that are reachable from v:

$$R(v) = \{u \in V : \text{ there is a path from } v \text{ to } u\}$$

Define $value(v)$ to be the minimum label in $R(v)$:

$$value(v) = min\{L(u) : u \in R(v)\}$$

Give an $\mathcal{O}(|V| + |E|)$-time algorithm that computes $value(v)$ for all vertices $v \in V$, that is, your algorithm must print $value(v)$ for each vertex v of G.

The graph is presented using the adjacency list data structure. So the input to your algorithm is a pair (n, Adj) where n is the number of vertices in the graph (we take $V = \{1, 2, \ldots, n\}$), and Adj is an array of length n whose element $Adj[v]$ is the (pointer to the head of the) linked list of neighbors of node v (for $1 \leq v \leq n$). If you need additional data structures (e.g., additional attributes associated with the vertices) clearly describe them.