Analysis of Algorithms: Solutions 6

Problem 1
Write a nonrecursive version of FIND-SET with path compression, for disjoint-set forests.

FIND-SET(x)

\[ z \leftarrow x \]
\[ \text{while } z \neq \text{parent}[z] \quad \triangleright \text{find the root} \]
\[ \quad \text{do } z \leftarrow \text{parent}[z] \]
\[ \text{while } x \neq z \quad \triangleright \text{set the parent pointers to the root } z \]
\[ \quad \text{do } y \leftarrow \text{parent}[x] \]
\[ \qquad \text{parent}[x] \leftarrow z \]
\[ \qquad x \leftarrow y \]
\[ \text{return } z \]
Problem 2
Write efficient algorithms for converting (a) an adjacency-list representation of a graph into an adjacency matrix and (b) an adjacency matrix into adjacency lists.

We denote the adjacency list of a vertex $u$ by $\text{Adj-List}[u]$, and the adjacency-matrix element for vertices $u$ and $v$ by $\text{Adj-Matrix}[u, v]$. The time complexity of both algorithms is $\Theta(V^2)$.

(a) Converting adjacency lists into a matrix.

$\text{LISTS-TO-MATRIX}(G) \quad G$ is represented by adjacency lists
for each $u \in V[G]
    \quad \text{do for each } v \in V[G]$
    \quad \quad \text{do } \text{Adj-Matrix}[u, v] \leftarrow 0$
for each $u \in V[G]$
    \quad \text{do for each } v \in \text{Adj-List}[u]$
    \quad \quad \text{do } \text{Adj-Matrix}[u, v] \leftarrow 1$

(b) Converting an adjacency matrix into lists.

$\text{MATRIX-TO-LISTS}(G) \quad G$ is represented by an adjacency matrix
for each $u \in V[G]$
    \quad \text{do initialize an empty list } \text{Adj-List}[u]$
for each $u \in V[G]$
    \quad \text{do for each } v \in V[G]$
    \quad \quad \text{do if } \text{Adj-Matrix}[u, v] = 1$
    \quad \quad \text{then add } v \text{ to } \text{Adj-List}[u]$

Problem 3
Using Figure 23.3 in the textbook as a model, illustrate the steps of breadth-first search on the directed graph of Figure 23.2(a), with vertex 3 as the source.

The order of painting the vertices is as follows:

\begin{align*}
\text{gray 3} & \quad \text{black 5} \\
\text{gray 5} & \quad \text{black 6} \\
\text{gray 6} & \quad \text{gray 2} \\
\text{black 3} & \quad \text{black 4} \\
\text{gray 4} & \quad \text{black 2}
\end{align*}