Exercise 1: Dijkstra’s Algorithm

Execute Dijkstra’s Algorithm on the following weighted, directed graph, starting at node $s$. Into the table further below, write the distances from each node to $s$ that the algorithm stores in the priority queue after each iteration.

![Graph Diagram]

<table>
<thead>
<tr>
<th>Initialization</th>
<th>$s$</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(s, \cdot)$</td>
<td>0</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

1. Step ($u = s$)  
   $\delta(s, \cdot) =$

2. Step ($u = $)  
   $\delta(s, \cdot) =$

3. Step ($u = $)  
   $\delta(s, \cdot) =$

4. Step ($u = $)  
   $\delta(s, \cdot) =$

5. Step ($u = $)  
   $\delta(s, \cdot) =$

6. Step ($u = $)  
   $\delta(s, \cdot) =$

7. Step ($u = $)  
   $\delta(s, \cdot) =$

8. Step ($u = $)  
   $\delta(s, \cdot) =$
Exercise 2: Currency Exchange

Consider $n$ currencies $w_1, \ldots, w_n$. The exchange rates are given in an $n \times n$-matrix $A$ with entries $a_{ij}$ ($i, j \in \{1, \ldots, n\}$). Entry $a_{ij}$ is the exchange rate from $w_i$ to $w_j$, i.e., for one unit of $w_i$ one gets $a_{ij}$ units of $w_j$.

Given a currency $w_{i_0}$, we want to find out whether there is a sequence $i_0, i_1, \ldots, i_k$ such that we make profit if we exchange one unit of $w_{i_0}$ to $w_{i_1}$, then to $w_{i_2}$ etc. until $w_{i_k}$ and then back to $w_{i_0}$.

(a) Translate this problem to a graph problem. That is, define a graph and a property which the graph fulfills if and only if there is a sequence of currencies as described above.

(b) Give an algorithm that decides in $O(n^3)$ time steps whether there is a sequence of currencies as described above. Explain the correctness and runtime.

*Hint:* $\log(ab) = \log a + \log b$. 