Conventions and Notation
Let $d$ be a document; we’ll use the following shorthand notation to describe the link structure surrounding $d$:

$\text{To}(d)$: the set of documents that link to $d$

$\text{From}(d)$: the set of documents that are linked to by $d$

Notice that the in-degree of $d$ is the number of documents in $\text{To}(d)$. We will be ignoring repeated links (i.e., if document $d$ has two hyperlinks to document $d'$, we will only count one link between them), and we will also ignore self-links (links from a document to itself).

Running Example: This graph shows the link structure between four documents W, X, Y, and Z:

```
    W -- Y -- Z
     \  |  /
      X
```

We have $\text{To}(W)$ consisting of just X, whereas $\text{To}(Y)$ is the two documents W and X. $\text{From}(X)$ is the two documents W and Y, and $\text{From}(Z)$ doesn’t contain any documents. Furthermore, note that the in-degree of W is the same as the in-degree of Z, and that the out-degrees of W and Y are equal.

Hubs and Authorities Algorithm
The algorithm processes queries in the following manner. First, we retrieve a root set of (hopefully) relevant documents via content-based IR. (One may expand this root set by adding in the documents that link to or are linked from some document in the root set.) Let $N$ be the number of documents in the root set, and for convenience let’s call these documents $d_1, d_2, \ldots, d_N$. For each $d_j$ in the root set, we want to compute its authority score $a_j$ and its hub score $h_j$.

1. Initialization: For every document $d_j$, set both $a_j$ and $h_j$ to 1.
2. Repeat the following steps in order until no changes occur:
   3. Update authority scores: For every document $d_j$, change $a_j$ to $\sum_{d_k \in \text{To}(d_j)} h_k$
   4. Pseudo-normalize\(^1\) authority scores: For every document $d_j$, change $a_j$ to $a_j / \sum_{k=1}^{N} a_k$
   5. Update hub scores: For every document $d_j$, change $h_j$ to $\sum_{d_k \in \text{From}(d_j)} a_k$
   6. Pseudo-normalize hub scores: For every document $d_j$, change $h_j$ to $h_j / \sum_{k=1}^{N} h_k$

\(^1\)We’re using pseudo-normalization rather than length-normalization to make the calculations a little easier.
Running Example, Computing Scores:

\[
\begin{array}{c}
W \rightarrow X \\
Y \rightarrow Z \\
Y \\
Z
\end{array}
\]

The following table computes the authority and hub scores for the four nodes in our example graph, for the first two iterations of the hubs and authorities algorithm:

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Init</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>b. Update-a</td>
<td>1 (1)</td>
<td>0 (1)</td>
<td>2 (1)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>c. Pnorm-a</td>
<td>1/4 (1)</td>
<td>0 (1)</td>
<td>1/2 (1)</td>
<td>1/4 (1)</td>
</tr>
<tr>
<td>d. Update-h</td>
<td>1/4 (1/2)</td>
<td>0 (3/4)</td>
<td>1/2 (1/4)</td>
<td>1/4 (0)</td>
</tr>
<tr>
<td>e. Pnorm-h</td>
<td>1/4 (1/3)</td>
<td>0 (1/2)</td>
<td>1/2 (1/6)</td>
<td>1/4 (0)</td>
</tr>
<tr>
<td>f. Update-a</td>
<td>1/2 (1/3)</td>
<td>0 (1/2)</td>
<td>5/6 (1/6)</td>
<td>1/6 (0)</td>
</tr>
<tr>
<td>g. Pnorm-a</td>
<td>1/3 (1/3)</td>
<td>0 (1/2)</td>
<td>5/9 (1/6)</td>
<td>1/9 (0)</td>
</tr>
<tr>
<td>h. Update-h</td>
<td>1/3 (5/9)</td>
<td>0 (8/9)</td>
<td>5/9 (1/9)</td>
<td>1/9 (0)</td>
</tr>
<tr>
<td>i. Update-h</td>
<td>1/3 (5/14)</td>
<td>0 (4/7)</td>
<td>5/9 (1/14)</td>
<td>1/9 (0)</td>
</tr>
</tbody>
</table>

Row e shows the result of running one iteration of the algorithm; row i shows the result of running two iterations of the algorithm.