Solutions to Practice Exercises

19.1 We do not consider the questions containing neither of the keywords as their relevance to the keywords is zero. The number of words in a question include stop words. We use the equations given in Section 19.2.1 to compute relevance; the log term in the equation is assumed to be to the base 2.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>84</td>
<td>1</td>
<td>0.0170</td>
<td>0.0170</td>
<td>0.0002</td>
<td>0.0004</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>0</td>
<td>0.0000</td>
<td>0.0641</td>
<td>0.0000</td>
<td>0.0029</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>1</td>
<td>0.0310</td>
<td>0.0310</td>
<td>0.0006</td>
<td>0.0006</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>1</td>
<td>0.0641</td>
<td>0.0000</td>
<td>0.0029</td>
<td>0.0029</td>
</tr>
<tr>
<td>7</td>
<td>33</td>
<td>1</td>
<td>0.0430</td>
<td>0.0430</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>1</td>
<td>0.0443</td>
<td>0.0129</td>
<td>0.0013</td>
<td>0.0040</td>
</tr>
<tr>
<td>9</td>
<td>77</td>
<td>0</td>
<td>0.0000</td>
<td>0.0186</td>
<td>0.0000</td>
<td>0.0002</td>
</tr>
<tr>
<td>14</td>
<td>30</td>
<td>1</td>
<td>0.0473</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0015</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
<td>1</td>
<td>0.0544</td>
<td>0.0544</td>
<td>0.0020</td>
<td>0.0041</td>
</tr>
</tbody>
</table>

19.2 Let \( S \) be a set of \( n \) keywords. An algorithm to find all documents that contain at least \( k \) of these keywords is given below:

This algorithm calculates a reference count for each document identifier. A reference count of \( i \) for a document identifier \( d \) means that at least \( i \) of the keywords in \( S \) occur in the document identified by \( d \). The algorithm maintains a
list of records, each having two fields – a document identifier, and the reference count for this identifier. This list is maintained sorted on the document identifier field.

initialize the list $L$ to the empty list;
for (each keyword $c$ in $S$) do
begin

$D :=$ the list of documents identifiers corresponding to $c$;
for (each document identifier $d$ in $D$) do
if (a record $R$ with document identifier as $d$ is on list $L$) then

$R.reference.count := R.reference.count + 1$;
else begin
make a new record $R$;
$R.document.id := d$;
$R.reference.count := 1$;
add $R$ to $L$;
end;
end;
for (each record $R$ in $L$) do
if ($R.reference.count >= k$) then
output $R$;

Note that execution of the second for statement causes the list $D$ to “merge” with the list $L$. Since the lists $L$ and $D$ are sorted, the time taken for this merge is proportional to the sum of the lengths of the two lists. Thus the algorithm runs in time (at most) proportional to $n$ times the sum total of the number of document identifiers corresponding to each keyword in $S$.

19.3 No answer
19.4 No answer
19.5 No answer