Question 3.1: Inverted files

Question: List the set of documents that satisfy the query
Obama *SLOP/2* **Election**

Solution: Documents 1 and 3

$$\textbf{Obama} = \langle 4 : \{1 \rightarrow [3]\}, \{2 \rightarrow [6]\}, \{3 \rightarrow [2, 17]\}, \{4 \rightarrow [1]\}\rangle$$

$$\textbf{Election} = \langle 4 : \{1 \rightarrow [1]\}, \{2 \rightarrow [1, 21]\}, \{3 \rightarrow [3]\}, \{5 \rightarrow [16, 22, 51]\}\rangle$$

Question 3.2: Inverted files

Question: List the values of x for which the query **Obama** *SLOP/x* **Election** has a different set of documents as answers (starting from x = 1).

Solution:

- **Obama** *SLOP/1* **Election** returns document 3
- **Obama** *SLOP/2* **Election** returns documents 3 and 1
- **Obama** *SLOP/5* **Election** returns documents 3,1, and 2
- Thus the values are x=1, x=2, and x=5

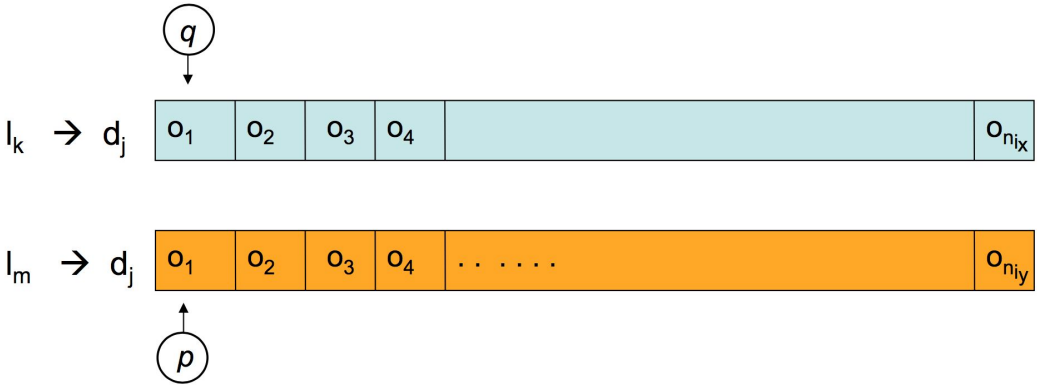
$$\mathbf{Obama} = \langle 4 : \{1 \rightarrow [3]\}, \{2 \rightarrow [6]\}, \{3 \rightarrow [2, 17]\}, \{4 \rightarrow [1]\}\rangle$$

$$\mathbf{Election} = \langle 4 : \{1 \rightarrow [1]\}, \{2 \rightarrow [1, 21]\}, \{3 \rightarrow [3]\}, \{5 \rightarrow [16, 22, 51]\}\rangle$$

Question 3.3: Inverted files

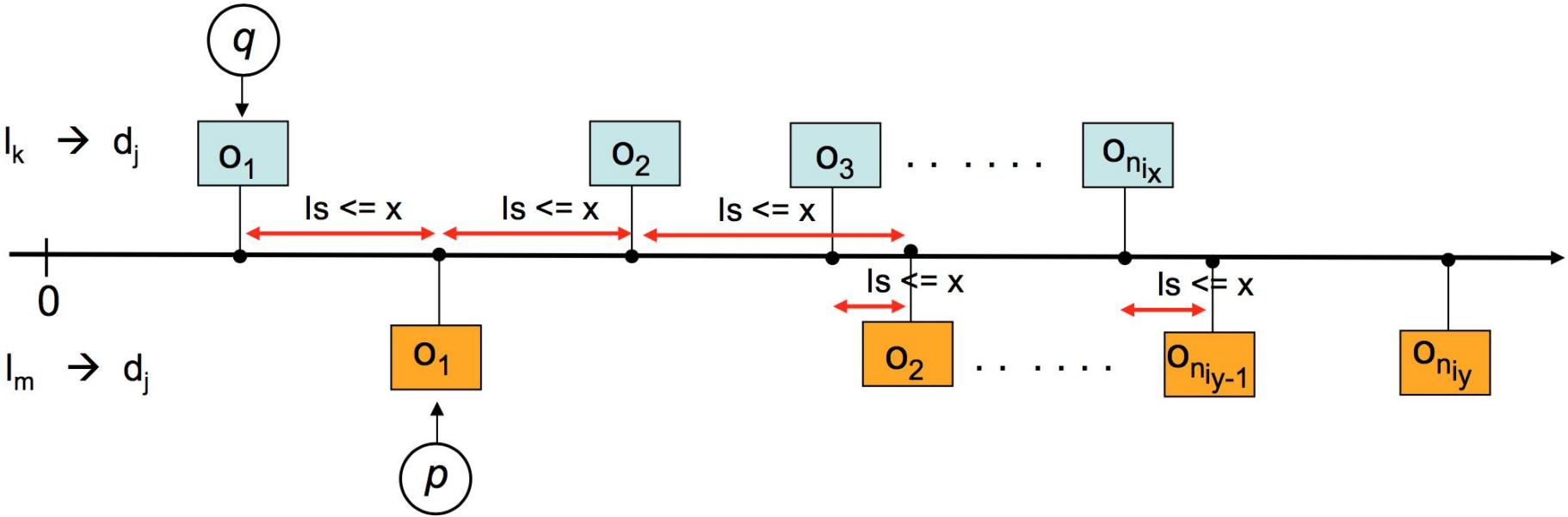
Solution: Correct choice is (i)

Justification:



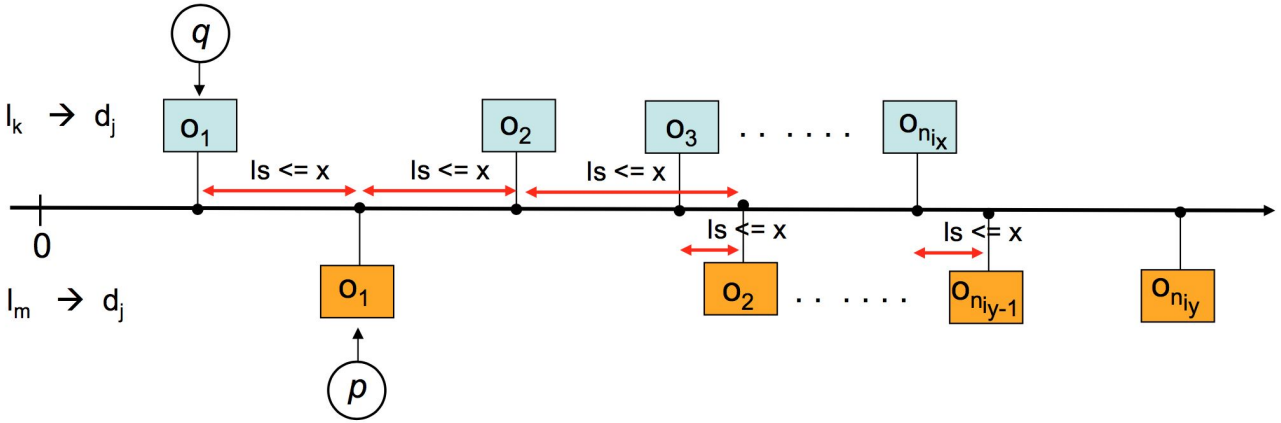
Total number of occurrences of the two terms in a document is *L*

Question 3.3: Inverted files



- Move the pointers as traditional merge: if val(q) < val(p): q = next(q); else: p = next(p)
- Check condition ls <= x after each move: terminate immediately if satisfied (hit)

Question 3.3: Inverted files



An example run (assume there is no hit):

Step 1. q = o1, p = o'1
Step 2. q = o2, p = o'1
Step 3. q = o2, p = o'2
Step 4. q = o3, p = o'2
Step 5. ...

Question 3.3: Inverted files

- Suppose *next(m)* gets the next location of pointer m, otherwise returns *null*
- Suppose *val(m)* gets the value pointed to by m.
- The merging algorithm is as follows

```
Input: p, q, and x

hit=0
while 1:
        If |val(p)-val(q)| <= x
             hit = 1
             break
        if val(p) <= val(q)
             p ← next(p)
        else
             q ← next(q)
        if (p==null) or (q==null)
              break
return (hit)
```