Information Retrieval

Weike Pan

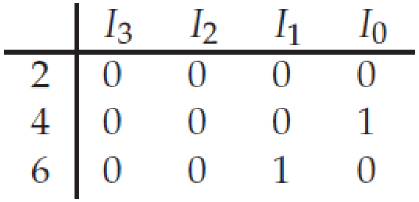
Exercise 4.2

How would you create the dictionary in blocked sort-based indexing on the fly to avoid an extra pass through the data?

ANSWER: Simply accumulate vocabulary in memory using, for example, a hash.

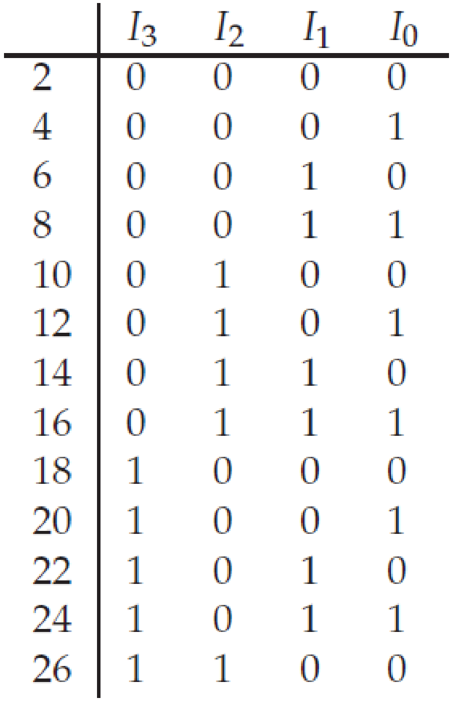
Exercise 4.4

For n=2 and 1≤T≤30, perform a step-by-step simulation of the algorithm in Figure 4.7. Create a table that shows, for each point in time at which T=2∗k tokens have been processed (1≤k≤15), which of the three indexes I0, . . . , I3 are in use. The first three lines of the table are given below.



	I_3	I_2	I_1	I_0
2	0	0	0	0
4	0	0	0	1
6	0	0	1	0

ANSWER:



	I_3	I_2	I_1	I_0
2	0	0	0	0
4	0	0	0	1
6	0	0	1	0
8	0	0	1	1
10	0	1	0	0
12	0	1	0	1
14	0	1	1	0
16	0	1	1	1
18	1	0	0	0
20	1	0	0	1
22	1	0	1	0
24	1	0	1	1
26	1	1	0	0

Exercise 4.9

Assume that machines in Map/Reduce have 100GB of disk space each. Assume further that the postings list of the term *the* has a size of 200GB. Then the Map/Reduce algorithm as described cannot be run to construct the index. How would you modify Map/Reduce so that it can handle this case?

ANSWER: Partition by docID as well as term for very frequent terms.

Exercise 4.11

Apply Map/Reduce to the problem of counting how often each term occurs in a set of files. Specify map and reduce operations for this task. Write down an example along the lines of Figure 4.6.

ANSWER:

map: input → list(word, 1)

reduce: (word,list(1)) → (word,length(list))