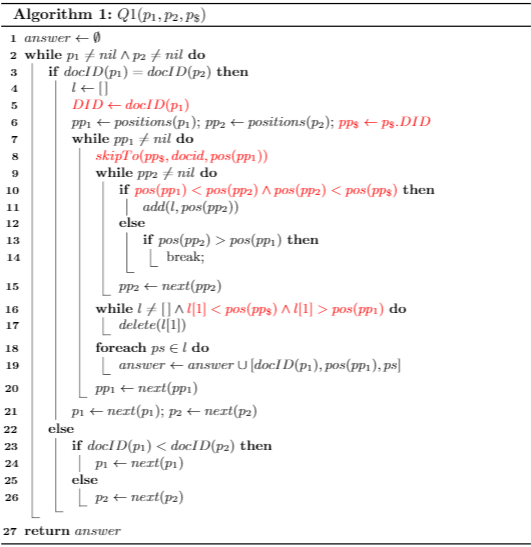
COMP6714 ASSIGNMENT 1

MINGLANG XIE z5228006

Q1



Q2

(1)

Assume that sub-indexes (each of pages) will be created if one chooses the no-merge strategy. Thus, there are pages in total, if the logarithmic merge strategy is used, we are following two rules:

1. only create and has size

2. whenever we have 2 , we need to merge it to form a

Therefore, when the generation grow to , we will get

when is sufficient large, . Therefore, if the logarithmic merge strategy is used, it will result in at most sub-indexes.

(2)

First, there are pages in total.

Second, we know the logarithmic merge strategy will result in at most , which means after generations, only the last generation in disk. Simply progress is:

Merge times

Merge times

…

Merge one time

Hence, the total cost of the logarithmic merge is:

The total I/O cost of the logarithmic merge is

Q3

After the encoding, the compressed non-positional inverted list is

Since this compressed list started with a , and is the only number to get a after the encoding. So, we can divide the compressed list as:

Then, we get the first document ID which is , and remian:

The next document ID will be , which is , because the remain compressed list started with , and so , and after , which give , use , we have , given , next, after given . Therefore, . Then remain:

Continue this progress, we can divide the compressed list as below:

Therefore, the document IDs in this list is .

Q4

The line 22 in Figure 2 that causes the bug, loop happens when pivot is unchanged, because *aterm.iterator.next(pivot)* did not move the cursor of the document ID, *findPivotTerm(terms, )* always return the same pivot.

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| UB | 3 | 5 | 4 |
| List |  |  |  |
|  |  |  |
|  |  |  |