Solution- Tutorial Set 1

1. **A. Term document matrix-**

	d1	d2	d3	d4
Approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
for	1	0	1	1
hopes	0	0	0	1
new	0	1	1	1
of	0	0	1	0
patients	0	0	0	1
schizophrenia	1	1	1	1
treatment	0	0	1	0

Inverted index-

Approach [1] : 3
breakthrough [1] : 1
drug [2] : 1->2
for [3]: 1->3- >4
hopes [1]: 4
new [3]: 2->3->4
of [1]: 3
patients [1]: 4
schizophrenia [4]: 1->2->3->4
treatment [1]: 3

B.

- schizophrenia AND drug - d1, d2
- for AND NOT(drug OR approach) - d4

2.

We can always intersect in O(qN) where q is the number of query terms and N the number of documents, so the intersection time is linear in the number of documents and query terms. Since the tightest bound for the size of the results list is N, the number of documents, you cannot do better than O(N).

3.

An efficient postings merge algorithm to evaluate x AND (NOT y) is:

a. MERGE(x, y, AND NOT)
b. answer<- ( )
c. while x!=NIL and y!=NIL
d. do if docID(x)=docID(y) then
    i. ADD(answer,docID(x))
    ii. x<- next(x)
    iii. y<-next(y)
e. else if docID(x)<docID(y)
        then ADD(answer,docID(x))
        x<- next(x)
f. else y<-next(y)
g. return(answer)

4.

i) using standard posting list:
All the entries (12) less than or equal to 81 from the first list will be compared. I.e. 12 Comparison

ii) Using postings list stored with skip pointers, with the suggested skip length of √P (P=length of the list).

[**2**, 4, 9, 12, **14**, 16, 18, 20, **24**, 32, 47, 81, **120**, 125, 158, 180]

2, 81
14, 81
24, 81
120, 81 (skip will not be occurred)
32, 81
4, 81
81, 81

5.

query -> Gates /k Microsoft
For k = 1:
Valid combination -> Gates (at position X) and Microsoft (at position X+-1)
Only doc 3 would satisfy (Gates at position 2 and Microsoft at position 3)

For k = 2:
Valid combination -> Gates (at position X) and Microsoft (at position X+-1, 2)
Doc1 and Doc3 satisfy this condition

6.

Consider two character strings s1 and s2. Without any loss in generality, let us assume that l1 = length(s1) < l2 = length(s2). For transforming s1 into s2,we can replace each character of s1 by the first l1 characters of s2 and insert the remaining characters of s2 at the end of s1.The number of operations incurred is l2 = max(l1, l2)

7.

SOLUTION.

		a	l	i	c	e
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**	**5** **5**
p	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**3** 4 / **3** **3**	**4** 5 / **4** **4**	**5** 6 / **5** **5**
a	**2** / **2**	**1** **2** / 3 **1**	**2** 3 / **2** **2**	**3** 4 / **3** **3**	**4** 5 / **4** **4**	**5** 6 / **5** **5**
r	**3** / **3**	3 **2** / 4 **2**	**2** **3** / 3 **2**	**3** 4 / **3** **3**	**4** 5 / **4** **4**	**5** 6 / **5** **5**
i	**4** / **4**	4 **3** / 5 **3**	**3** **3** / 4 **3**	**2** **4** / 4 **2**	4 5 / **3** **3**	5 6 / **4** **4**
s	**5** / **5**	5 **4** / 6 **4**	**4** **4** / 5 **4**	4 **3** / 5 **3**	**3** **4** / 4 **3**	**4** 5 / **4** **4**