Assignment1

Full name: Rong Zhen **zid: z5225226**

Q1:(1)

```
def Intersect(A,B):
    if A is equal to [] OR B is equal to []:
        return []
    else if A.Len is equal to 1 and B.Len is equal to 1:
        if A is equal to B:
            return A
        else:
            return []
    else:
        A_left = A[:A.Len/2]
        A_right = A[A.Len/2:]
        B_left = B[:B.Len/2]
        B_right = B[B.Len/2:]
        result = Intersect(A_left,B_left)+Intersect(A_right,B_left)
                +Intersect(A_left,B_right)+Intersect(A_right,B_right)
        return result
```

(2)

```
def divideToSublist(A,B):
    A1,B1 = newFunction(A,B,k)
    return A1,B1
def newFunction(A,B,new_k):
    if new_k<2:
        return [A],[B]
    else:
        A_left = A[:A.Len/2]
        A_right = A[A.Len/2:]
        B_left = B[:B.Len/2]
        B_right = B[B.Len/2:]
        A1,B1 = newFunction(A_left,B_left,floor(new_k/2))
        A2,B2 = newFunction(A_right,B_right,new_k-floor(new_k/2))
        return A1+A2,B1+B2
```

In this problem, everytIme the list is divided into 2 sub-list. So k will decreased to k/2 everytime. As known, k>=2. So we return the list of sub-list of each input when k<2.

Q2:(1)

As known, t sub-indexes(each of M pages) will be created if one chooses the no-merge strategy. So the collection size is t*M. And using the process of Logarithmic merge, I can get a table like below:

Level 0	M
Level1	2 M
Level2	4 M
……	……
Level h	2^h M

The worst situation is that each level has one sub-index, which means

$$M + 2M + 4M + \cdots + 2^h M = t * M$$
$$M * (1 + 2 + 4 + \cdots 2^h) = t * M$$
$$\frac{1 * (2^h - 1)}{2 - 1} = t$$
$$h = \log_2(t + 1)$$

when t is large, 1 can be ignored.

So using Logarithmic merge, it will result in at most $\lceil \log_2 t \rceil$.

(2) Named indexes in each Level as $I_0, I_1, \ldots\ldots, I_h$. The number of I_h is 1 and two I_{h-1} merge once can get the I_h. So I can get a table like below.

$$I_h \xrightarrow[merge\ once]{} I_{h-1} = 2 * I_{h-2} = 4 * I_{h-3} = \cdots = 2^{h-1} * I_0$$

$$I_{h-1} \xrightarrow[merge\ once]{} I_{h-2} = 2 * I_{h-3} = 4 * I_{h-4} = \cdots = 2^{h-2} * I_0$$

$$\vdots$$

$$I_1 \xrightarrow[merge\ once]{} 2^0 * I_0$$

And the number of indexes is different, like the table below:

I_h	1
I_{h-1}	2
I_{h-2}	4
I_{h-3}	8
……	
I_1	2^h

$$cost\ of\ merge = (2^0 * 2^{h-1} + 2^1 * 2^{h-2} + \cdots + 2^{h-1} * 2^0) * M$$
$$= (h * 2^{h-1}) * M = M * \log_2 t * 2^{\log_2 t - 1} = t * M * \log_2 t\ (1\ can\ be\ ignored)$$
$$cost = cost_{read\ index} + cost_{merge} = M * t + t * M * \log_2 t = O(t * M * \log_2 t)$$

Q3

(1) $Precision = \frac{6}{20} = 0.3$

(2) $F1 = \frac{2*0.3*\frac{6}{8}}{\left(\frac{6}{20}+\frac{6}{8}\right)} = 0.4286$

(3)

	1	2	3	4	5	6	7	8	9	10
Precision	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{2}{6}$	$\frac{2}{7}$	$\frac{2}{8}$	$\frac{3}{9}$	$\frac{3}{10}$
Recall	0.125	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.375	0.375

So the answer is 100%, 66.67%, 50%, 40%, 33.33%, 28.57%, 25%.

(4)

	11	12	13	14	15	16	17	18	19	20
Precision	$\frac{4}{11}$	$\frac{4}{12}$	$\frac{4}{13}$	$\frac{4}{14}$	$\frac{5}{15}$	$\frac{5}{16}$	$\frac{5}{17}$	$\frac{5}{18}$	$\frac{5}{19}$	$\frac{6}{20}$
Recall	0.5	0.5	0.5	0.5	0.625	0.625	0.625	0.625	0.625	0.75

The maximun precision value after 33% recall is $\frac{4}{11}$ = 0.3636

(5)

MAP = $\frac{1}{8} * \left(\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20}\right) = 0.4163$

(6) Assume the 21st and 22nd is relevant.

MAP = $\frac{1}{8} * \left(\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{21} + \frac{8}{22}\right) = 0.5034$

(7) Assume the 9999th and 10000th is relevant

MAP = $\frac{1}{8} * \left(\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{9999} + \frac{8}{10000}\right) = 0.4165$

(8) 0.5034-0.4163 = 0.0871

Q4

(1) Answer:

$$p(Q|d_1) = \prod \frac{tf}{\#of\ tokens\ in\ d1} = \frac{2}{10} * \frac{3}{10} * \frac{1}{10} * \ldots * 0 = 0$$

$$p(Q|d_2) = \prod \frac{tf}{\#\ of\ tokens\ in\ d2} = \frac{7}{10} * \frac{1}{10} * \ldots * 0 = 0$$

These two documents are same.

(2) Answer:

$$p(w|d_1) = \left(\frac{2}{10} * 0.8 + 0.2 * 0.8\right) * \left(\frac{3}{10} * 0.8 + 0.2 * 0.1\right) * \ldots * \left(\frac{0}{10} * 0.8 + 0.2 * 0.025\right)$$
$$= 0.000000962676$$

$$p(w|d_2) = \left(\frac{7}{10} * 0.8 + 0.2 * 0.8\right) * \left(\frac{1}{10} * 0.8 + 0.2 * 0.1\right) * \ldots * \left(\frac{0}{10} * 0.8 + 0.2 * 0.025\right)$$
$$= 0.000000013005$$

Document 1 would be ranked higher.