QUIZ-4 SOLUTIONS

BUCKETS	HASH FUNC 1 (i+j mod 3) ①	HASH FUNC 2 (j mod 3) ②	[1.5] Counts	[1.5] ~~Counts~~	Bitmap	
			①	②	①	②
0	(1,2) (1,5) (2,4) (4,5)	(1,3) (1,3) (2,3) (2,3)	4	4	1	1
1	(1,3) (3,4) (1,3) (3,4)	(2,4) (3,4) (3,4)	4	3	1	1
2	(2,3) (3,5) (2,3) (3,5)	(1,2) (3,5) (3,5) (4,5) (1,5)	4	5	1	1

Multihash

1st PASS [0.75]

item counts:
1 => 2 , 2 => 2 , 3 => 4 , 4 => 2 , 5 => 2

Buckets	Hash ①	Hash ②
0	4	4
1	4	3
2	4	5

2ND PASS [0.75]

Freq items: 1,2,3,4,5

BITMAP ① :

B0	B1	B2
1	1	1

BITMAP ② :

B0	B1	B2
1	1	1

Candidate item pairs :

{1,2} {1,3} {1,5} {2,3} {2,4}
{3,4} {3,5} {4,5}

Counts of pairs

{1,2} => 1 {1,3} => 2 {1,5} => 1
{2,3} => 2 {2,3} => 1 {3,4} => 2
{3,5} => 2 {4,5} => 1

· Frequent Pairs :- {1,3} , {2,3} {3,4} , {3,5}

MULTI STAGE

1st PASS [0.5 POINTS]

item counts :

1 ⇒ 2 , 2 => 2 , 3 => 4 , 4 => 2 , 5 => 2

HASH ①

BUCKET	COUNT
0	4
1	4
2	4

2ND PASS [0.5 POINTS]

frequent items : 1, 2, 3, 4, 5

BITMAP ① :

B0	B1	B2
1	1	1

HASH ②

BUCKET	COUNT
0	4
1	3
2	5

3RD PASS [0.5 POINTS]

frequent items : 1, 2, 3, 4, 5

BITMAP ②

B0	B1	B2
1	1	1

Count of pairs :

{1,2} => 1 , {1,3} = 2 , {1,5} => 1 , {2,3} => 2 , {2,4} => 1 ,
{3,4} => 2 , {3,5} => 2 , {4,5} = 1

Frequent Pairs : {1,3}, {2,3} , {3,4}, {3,5}

2) General idea - [0.5]

For some applications, it is sufficient to discover most frequent itemsets and is not essential ~~every~~ to discover every single one.

Pros: less I/o cost, time, passes. [0.25]

Cons: False positive, False Negative are induced in the results. [0.25]

3) PHASE 1 MAP i/p :-

[0.25] a chunk/subset of all baskets (sample of i/p file)

PHASE 1 MAP O/P :-

set of pairs (F,1) where F is a frequent itemset from sample.

PHASE 1 REDUCE I/P :-

[0.25] set of pairs (F,1)

PHASE 1 REDUCE O/P :-

Candidate itemsets

PHASE 2 MAP i/P :-

[0.25] Result from phase1 and total output file

PHASE 2 MAP O/P :-

Set of pairs (C,V), C is candidate itemsets, V is the support for that itemset.

PHASE 2 Reduce i/p

[0.25] (C,V)

PHASE 2 REDUCE O/P :-

if V≥S, emit (C,V)

4) False positives :- Infrequents in entire data, frequent in sample. [0.25]

False Negative :- Frequent in entire data, Infrequent in sample. [0.25]

Increasing Support: will increase FN as it will be harder to be frequent in the sample, decrease FP.

[0.5] Decreasing Support: will induce more FP in the data as it is easier to be frequent in the data/sample decrease FNs.

5)

[0.25] Singleton : {a} is in -ve border,

iff {a} is not frequent in the sample

[0.25] Pair : {a,b} is in -ve border,

iff {a,b} is not frequent in the sample

{a},{b} are frequent

0.5 marks

① Construct sample data set

② Find candidates frequent itemsets from sample

③ Construct Negative border.

④ process the whole file,

if no itemset from -ve border turns out to be frequent in whole dataset, correct ✓

if some ...., Repeat the algo with random Sample.