Quiz 3

- There are 6 problems total worth 35 points as shown in each question.
- You must not communicate with other students during this test.
- No books, notes allowed.
- No other electronic device except calculators are allowed. You cannot use your mobile as calculators.
- This is a 45 minute exam.
- Do not turn this page until instructed to.
- There are several different versions of this exam.

1. Fill in your information:

Full Name: ______________________________

NetID: ______________________________

1/1. (5 points) Suppose we use Bottom-Up Computation (BUC) to materialize cubes. We have a 3-D data array containing three dimensions A, B, C. The data contained in the array is as follows:

$(a_0; b_0; c_0) : 1$	$(a_0; b_0; c_1) : 1$	$(a_0; b_0; c_2) : 1$
$(a_0; b_1; c_0) : 1$	$(a_0; b_1; c_1) : 1$	$(a_0; b_1; c_2) : 1$
$(a_0; b_2; c_0) : 1$	$(a_0; b_2; c_1) : 1$	$(a_0; b_2; c_2) : 1$
$(a_0; b_3; c_0) : 1$	$(a_0; b_3; c_1) : 1$	$(a_0; b_3; c_2) : 1$

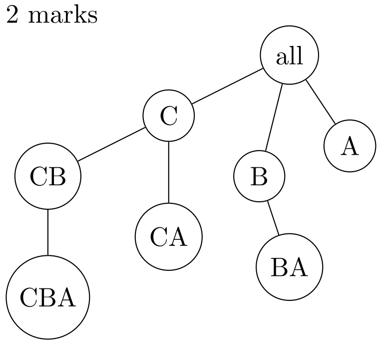
Now suppose we construct an iceberg cube for dimension A, B, C with different orders of exploration.

A. Draw the trace trees of expansion with regard to exploration order: C, B, A.

B. Suppose the minimum support = 4 with the exploration order of A, B, C, how many cells would be computed? Please give detailed explanation.

Solution.

A. 2 marks



B. 16 Cubes to be computed, $All, a_0, a_0b_0, a_0b_1, a_0b_2, a_0b_3, a_0c_0, a_0c_1, a_0c_2, b_0, b_1, b_2, b_3, c_0, c_1, c_2$ - 3 marks (Partial marks for partial answer)

1/2. (5 points) Suppose we use Bottom-Up Computation (BUC) to materialize cubes. We have a 3-D data array containing three dimensions A, B, C. The data contained in the array is as follows:

$(a_0; b_0; c_0) : 1$	$(a_0; b_0; c_1) : 1$	$(a_0; b_0; c_2) : 1$
$(a_0; b_1; c_0) : 2$	$(a_0; b_1; c_1) : 2$	$(a_0; b_1; c_2) : 2$
$(a_0; b_2; c_0) : 1$	$(a_0; b_2; c_1) : 1$	$(a_0; b_2; c_2) : 1$
$(a_0; b_3; c_0) : 1$	$(a_0; b_3; c_1) : 1$	$(a_0; b_3; c_2) : 1$

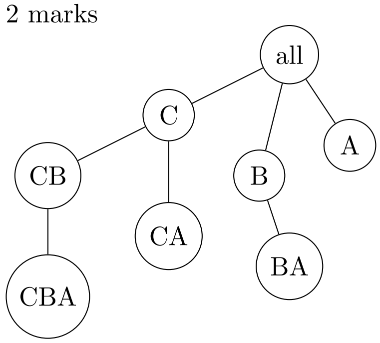
Now suppose we construct an iceberg cube for dimension A, B, C with different orders of exploration.

A. Draw the trace trees of expansion with regard to exploration order: C, B, A.

B. Suppose the minimum support = 4 with the exploration order of A, B, C, how many cells would be computed? Please give detailed explanation.

Solution.

A. 2 marks



B. 22 Cubes to be computed, -3 marks (Partial marks for partial answer)

1/3. (5 points) Suppose we use Bottom-Up Computation (BUC) to materialize cubes. We have a 3-D data array containing three dimensions A, B, C. The data contained in the array is as follows:

$(a_0; b_0; c_0) : 1$	$(a_0; b_0; c_1) : 1$	$(a_0; b_0; c_2) : 3$
$(a_0; b_1; c_0) : 1$	$(a_0; b_1; c_1) : 1$	$(a_0; b_1; c_2) : 1$
$(a_0; b_2; c_0) : 2$	$(a_0; b_2; c_1) : 1$	$(a_0; b_2; c_2) : 1$
$(a_0; b_3; c_0) : 1$	$(a_0; b_3; c_1) : 2$	$(a_0; b_3; c_2) : 1$

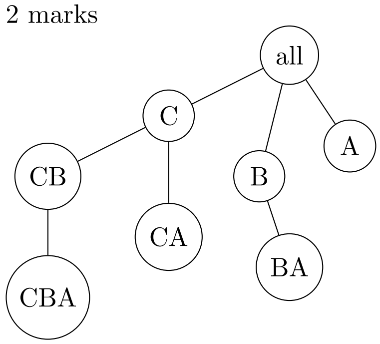
Now suppose we construct an iceberg cube for dimension A, B, C with different orders of exploration.

A. Draw the trace trees of expansion with regard to exploration order: C, B, A.

B. Suppose the minimum support = 4 with the exploration order of A, B, C, how many cells would be computed? Please give detailed explanation.

Solution.

A. 2 marks



B. 31 Cubes to be computed, - 3 marks (Partial marks for partial answer)

2/1. (5 points)

Suppose the standard deviation of n observations $x_1, x_2......x_n$ as

$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$

Is this measure a distributive, algebraic or holistic measure? Justify your answer.

Solution. Algebraic - 2 marks

After some calculations, $s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2 - \frac{1}{N}(x_i{}^2)}$. Algebraic since the measure can be computed based on three distributive measures: n, $\sum x_i{}^2$ and $\sum x_i$ - 3 marks

---

3/1. (5 points) Given a database of T tuples, D dimensions, and F shell fragment size, the fragment cubes's space requirement is:

$$O(T\lceil \frac{D}{F} \rceil(2^F - 1))$$

Consider two cases, where F = 1 and F = D,

A. In what case is the space requirement lowest?

B. What are challenges in using this value of F?

Solution.

A. F =1 O(TD), F = D $O(T(2^D - 1))$, 2 marks Space lowest on F = 1 - 1 Mark

B. Long query execution time - 2 marks

4/1. (5 points) We have a data array containing 3 dimensions A, B and C. The 3-D array is divided into small chunks. Each dimension is divided into 3 equally sized partitions. See Figure 1. For example, dimension A is divided into a_0, a_1 and a_2, and dimension B is divided into b_0, b_1 and b_2. There are totally 27 chunks and each chunk is represented by a sub-cube $a_i b_j c_k$. The cardinality (size) of the dimensions A, B, and C is 900, 300, and 600. Since we divide each dimension into 3 parts with equal size, the sizes of the chunks on dimensions A, B, and C are 300, 100, and 200 respectively. Now we want to use **Multiway Array Aggregation** Computation to materialize cubes. The base cuboid ABC is computed as a 3-D array. We want to materialize the 2-D cuboids AB, AC and BC.

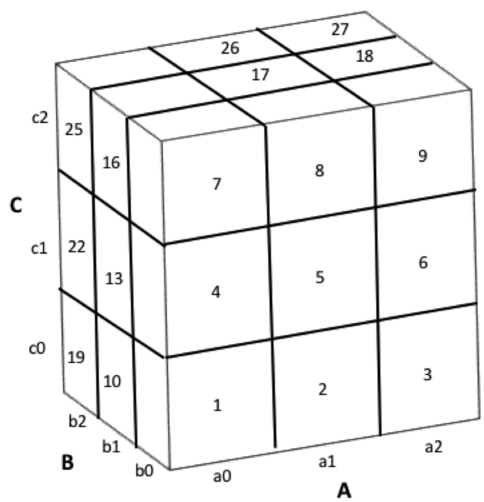


Figure 1: Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

If we scan the chunk in the order $1, 2, 3, ...27$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading a 3-D chunk into memory repeatedly, what is the minimum memory requirement to hold all the related 2-D planes?

Solution. 5 marks (Partial marks if almost right)

Space requirement = AC + AB + BC = 900*600 + 100*900 + 100*200 = 650,000 - 5 marks

---

4/2. (5 points) We have a data array containing 3 dimensions A, B and C. The 3-D array is divided into small chunks. Each dimension is divided into 3 equally sized partitions. See Figure 1. For example, dimension A is divided into a_0, a_1 and a_2, and dimension B is divided into b_0, b_1 and b_2. There are totally 27 chunks and each chunk is represented by a sub-cube $a_i b_j c_k$. The cardinality (size) of the dimensions A, B, and C is 300, 900, and 600. Since we divide each dimension into 3 parts with equal size, the sizes of the chunks on dimensions A, B, and C are 100, 300, and 200 respectively. Now we want to use **Multiway Array Aggregation** Computation to materialize cubes. The base cuboid ABC is computed as a 3-D array. We want to materialize the 2-D cuboids AB, AC and BC.

If we scan the chunk in the order $1, 2, 3, ...27$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading a 3-D chunk into memory repeatedly, what is the minimum memory requirement to hold all the related 2-D planes?

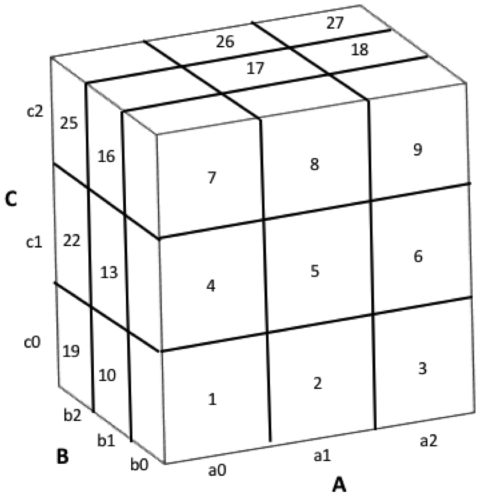


Figure 2: Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

Solution. 5 marks (Partial marks if almost right)
Space requirement = AC + AB + BC = 300*600 + 300*300 + 300*200 = 330,000 - 5 marks

---

4/3. (5 points) We have a data array containing 3 dimensions A, B and C. The 3-D array is divided into small chunks. Each dimension is divided into 3 equally sized partitions. See Figure 1. For example, dimension A is divided into a_0, a_1 and a_2, and dimension B is divided into b_0, b_1 and b_2. There are totally 27 chunks and each chunk is represented by a sub-cube $a_i b_j c_k$. The cardinality (size) of the dimensions A, B, and C is 600, 300, and 900. Since we divide each dimension into 3 parts with equal size, the sizes of the chunks on dimensions A, B, and C are 200, 100, and 300 respectively. Now we want to use **Multiway Array Aggregation** Computation to materialize cubes. The base cuboid ABC is computed as a 3-D array. We want to materialize the 2-D cuboids AB, AC and BC.

If we scan the chunk in the order $1, 2, 3, ...27$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading a 3-D chunk into memory repeatedly, what is the minimum memory requirement to hold all the related 2-D planes?

Solution. 5 marks (Partial marks if almost right)
Space requirement = AC + AB + BC = 600*900 + 600*100 + 100*300 = 630,000 - 5 marks

---

4/4. (5 points) We have a data array containing 3 dimensions A, B and C. The 3-D array is divided into small chunks. Each dimension is divided into 3 equally sized partitions. See Figure 1. For example, dimension A is divided into a_0, a_1 and a_2, and dimension B is divided into b_0, b_1 and b_2. There are totally 27 chunks and each chunk is represented by a sub-cube $a_i b_j c_k$. The cardinality (size) of the dimensions A, B, and C is 300, 600, and 900. Since we divide each dimension into 3

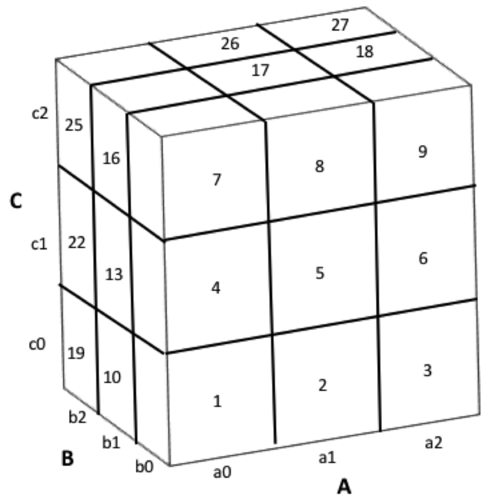


Figure 3: Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

parts with equal size, the sizes of the chunks on dimensions A, B, and C are 100, 200, and 300 respectively. Now we want to use **Multiway Array Aggregation** Computation to materialize cubes. The base cuboid ABC is computed as a 3-D array. We want to materialize the 2-D cuboids AB, AC and BC.

If we scan the chunk in the order $1, 2, 3, ...27$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading a 3-D chunk into memory repeatedly, what is the minimum memory requirement to hold all the related 2-D planes?

Solution. 5 marks (Partial marks if almost right)
Space requirement = AC + AB + BC = 300*900 + 300*200 + 200*300 = 390,000 - 5 marks

---

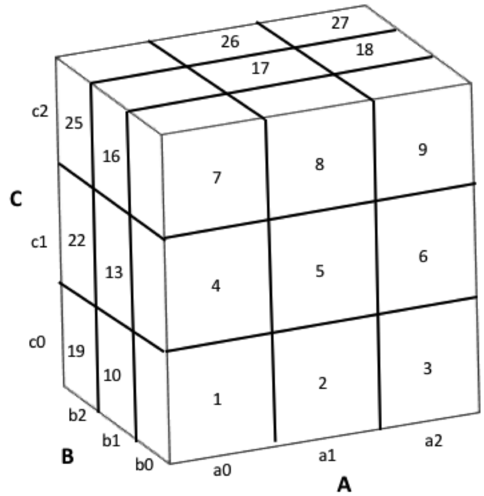


Figure 4: Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

5/1. (5 points) Given the following database with $minsup = 0.6$.

TransactionID	Items
T100	K,A,D,B,C
T200	D,A,E,F
T300	C,D,B,E
T400	B,A,C,K,D
T500	A,G,C

A. Generate an ordered list of frequent items based on the raw transaction database.
Hint: Reorder items within each transaction according to their frequencies in the whole database.

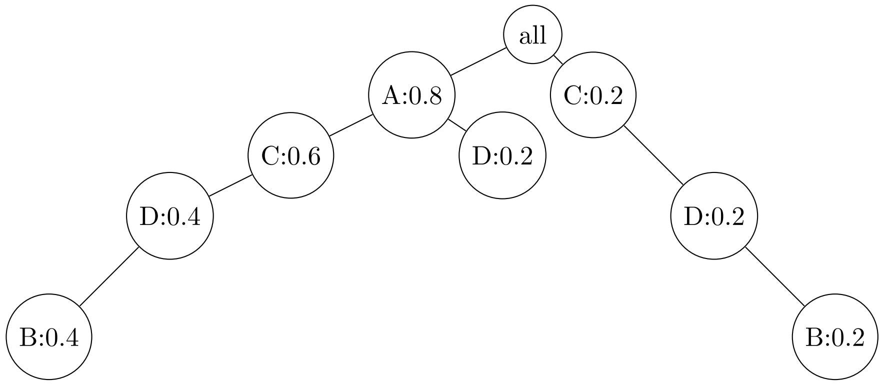
B. Generate FP-tree of this transaction database based on the frequent item list.

Solution.

A. 2 marks (Don't care about order)

TransactionID	Items
T100	A,D,C,B
T200	D,A
T300	C,D,B
T400	A,C,D,B
T500	A,C

B. 3 marks



5/2. (5 points) Given the following database with $minsup = 0.4$.

TransactionID	Items
T100	A,B,C,D
T200	A,B,C,E
T300	A,B,E,F,H
T400	A,C,H

A. Generate an ordered list of frequent items based on the raw transaction database.
Hint: Reorder items within each transaction according to their frequencies in the whole database.

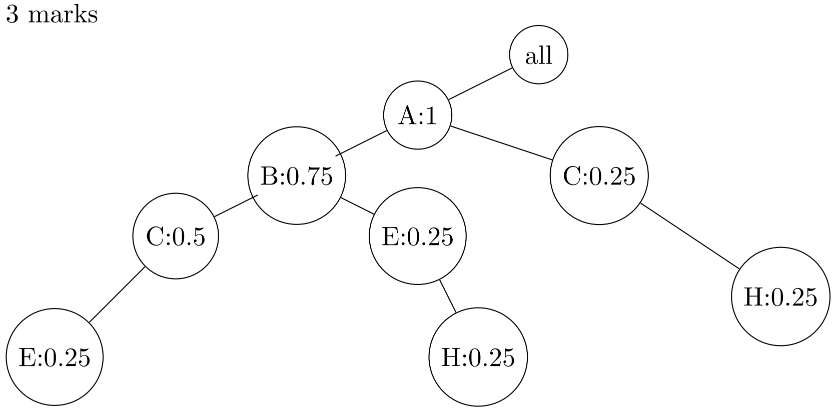
B. Generate FP-tree of this transaction database based on the frequent item list.

Solution.

A. 2 marks(Don't care about order)

TransactionID	Items
T100	A,B,C
T200	A,B,C,E
T300	A,B,E,H
T400	A,C,H

B. 3 marks



5/3. (5 points) Given the following database with $minsup = 0.4$.

TransactionID	Items
T100	K,A,D,B,C,E
T200	D,A,E,F
T300	K,C,D,B,E
T400	B,A,C,K,E
T500	A,G,C

A. Generate an ordered list of frequent items based on the raw transaction database.
Hint: Reorder items within each transaction according to their frequencies in the whole database.

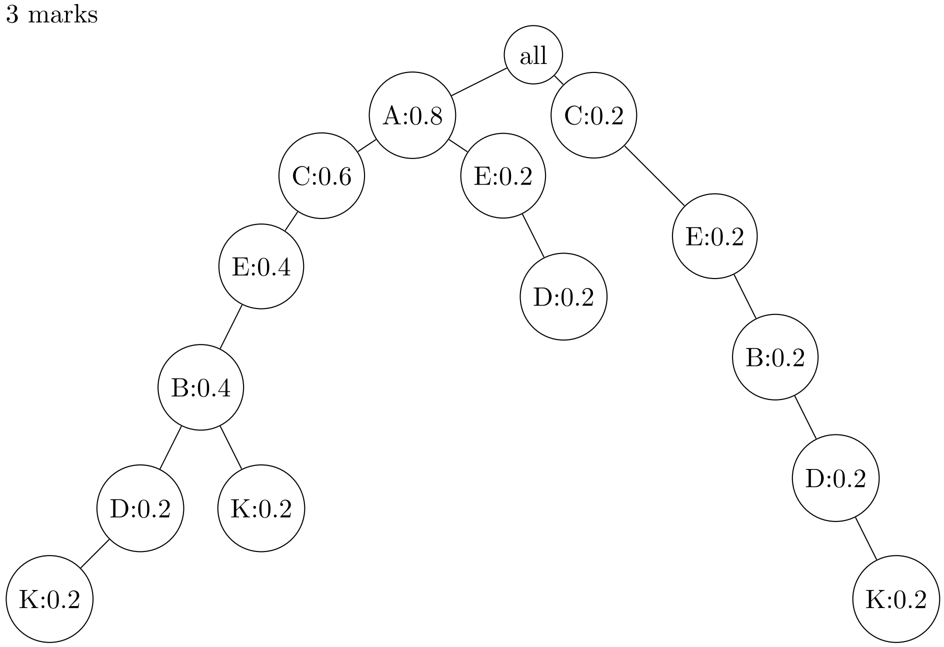
B. Generate FP-tree of this transaction database based on the frequent item list.

Solution.

A. 2 marks

TransactionID	Items
T100	K,A,D,B,C,E
T200	D,A,E
T300	K,C,D,B,E
T400	B,A,C,K,E
T500	A,C

B. 3 marks



6/1. (10 points) Consider the following transaction database:

TransactionID	Items
T1	A,B,C,D
T2	A,B,C,E
T3	A,B,E,F,H
T4	A,C,H

Suppose that minimum support is set to 50% and minimum confidence to 60%.

A. List all frequent item sets together with their support.

B. Which of the item sets from A) are closed? Which of the item sets from A) are maximal?

C. List at least 2 association rules with $minconf = 0.6$ from the frequent maximal patterns computed in part B.

Solution.

A. 4 marks (Partial marks if missed some rules)
$A = 1, B = 0.75, C = 0.75, E = 0.5, H = 0.5,$
AB = 0.75, AC = 0.75, AE = 0.5, AH = 0.5, BC = 0.5, BE = 0.5
ABC = 0.5 , ABE = 0.5

B. 4 marks (2 for closed, 2 for maximal) Partial marks if missed some rules
Closed : A,AB, AC, AH, ABC, ABE
Maximal : AH, ABC,ABE

C. 2 marks for any two rules
ABC
$AB- > C$ 0.66, $AC- > B$ 0.66, $BC- > A$ 1, $B- > AC$ 0.66, $C- > AB$ 0.66
ABE
$AB- > E$ 0.66, $AE- > B$ 1, $BE- > A$ 1, $B- > AE$ 0.66, $E- > AB$ 1
AH
$H- > A$ 1