

Exercice 1

Soit \mathcal{D}_1 une base de données au format transactionnel et avec la représentation horizontale $\mathcal{H}_{\mathcal{D}_1}$ suivante :

trans.	Items					
t_1		B	C	D		
t_2	A	B	C		E	
t_3	A	B	C	D		F
t_4				D	E	
t_5	A	B				
t_6	A		C		E	F
t_7	A	B			E	F
t_8				D		F
t_9			C		E	
t_{10}	A	B				F

Question 1 • Donnez la représentation verticale $\mathcal{V}_{\mathcal{D}_1}$ et la représentation matricielle $\mathcal{M}_{\mathcal{D}_1}$ de \mathcal{D}_1 .

Question 2 • calculez le support, la fréquence (relative/absolue) des itemsets suivants :

$$L = \{ACD, CE, BCE, ABCE, E, D, BC, F, CDF, EF\}$$

Question 3 • Calculez les itemsets fréquents avec un minimum support $\theta \in \{5, 6, 7, 8, 9, 10\}$

Question 4 • Donnez un exemple de deux itemsets comparables et de deux itemsets non comparables.

Exercice 2

Question 1 • Écrire une preuve pour la propriété anti-monotone des itemsets fréquents.

Question 2 • Écrire une preuve pour la propriété Apriori.

Exercice 3

Soit \mathcal{D}_2 une base de données au format transactionnel :

t	Items			
t_1	A	C	D	
t_2	B	C	E	
t_3	A	B	C	E
t_4	B		E	
t_5	A	B	C	E
t_6	B	C	E	

- Question 1** • Déroulez l'algorithme Apriori sur \mathcal{D}_2 avec un minsup $\theta = 3$ et sans l'opérateur canonique κ .
- Question 2** • Déroulez l'algorithme Apriori sur \mathcal{D}_2 avec un minsup $\theta = 3$ avec l'opérateur child basé sur un ordre lexicographique lex.
- Question 3** • Implémentez en Java l'algorithme Apriori avec et sans l'opérateur child+lex. Comparez les deux versions sur les bases fournies dans `.\DataSets\`.
- Question 4** • Proposez un algorithme et une exploration bottom-up pour extraire l'ensemble des itemsets fréquents. Codez-le et comparez le avec la version Apriori.
- Question 5** • Révisez l'algorithme Apriori de sorte à pouvoir extraire uniquement les itemsets fréquents qui ont une taille supérieure à une valeur *size*. Codez cette nouvelle version.