

Bases de Données II, Charleroi, 6 janvier 2025

NOM + PRÉNOM :

Orientation + Année :

Cet examen contient 7 questions. Durée : exactement 2 heures et 50 minutes. Les questions sont censées être claires. Aucune clarification supplémentaire ne sera fournie pendant l'examen. Si une question vous semble ambiguë ou incomplète, veuillez formuler vos hypothèses et répondez en fonction de celles-ci. **Il est permis de détacher la dernière page.**

Un fleuriste livre des bouquets à la maison. Chaque espèce de fleur (tulipe, rose...) a un prix exprimé en centimes d'euro. Par exemple, le prix d'une rose est de 150 centimes d'euro, indépendamment de sa couleur ; voir la ligne

```
<fleur fnom="rose" prix="150"/>
```

Un bouquet rassemble des fleurs de différentes espèces et couleurs. Par exemple, le bouquet **Exotique** est composé d'un seul tournesol et trois iris bleus. Voir les lignes :

```
<bouquet bnom="Exotique">
  <fleur fnom="tournesol" couleur="jaune" nombre="1"/>
  <fleur fnom="iris" couleur="bleu" nombre="3"/>
</bouquet>
```

La figure 1 montre le document XML et son DTD.

Question 1 Rédigez une requête en **XPath** qui renvoie le nom de chaque bouquet contenant des fleurs d'espèces différentes partageant la même couleur. L'utilisation des fonctions d'agrégation, telles que `count`, est interdite. Évitez également l'utilisation des axes `parent` et `ancestor`.

Pour le document XML de la figure 1, la réponse est la suivante :

```
bnom="Printemps"
```

En effet, le bouquet **Printemps** fait partie de la réponse, car il contient une rose et une tulipe ayant la même couleur, **jaune**. Notez que le bouquet **Belge** ne figure pas dans le résultat. **Assurez-vous que les crochets et les parenthèses sont correctement imbriqués.**

.../5

```
//bouquet[fleur[@couleur=following-sibling::fleur/@couleur]]/@bnom
```

Question 2 Écrivez une requête en **XPath** qui renvoie le nom du bouquet contenant le plus grand nombre de tulipes. La fonction **sum** est la seule fonction d'agrégation autorisée. L'utilisation de **max** et **min** est interdite.

Pour le document XML de la figure 1, la réponse est la suivante :

```
bnom="Belge"
```

Notez que le bouquet **Belge** contient 13 tulipes, et qu'aucun autre bouquet ne contient autant de tulipes. **Assurez-vous que les crochets et les parenthèses sont correctement imbriqués.**

.../5

```
//bouquet[not(sum(fleur[@fnom="tulipe"]/@nombre)<>//bouquet/sum(fleur[@fnom="tulipe"]/@nombre))]/@bnom
```

Question 3 Le *colorama* d'une espèce de fleur dans un bouquet est l'ensemble des couleurs des fleurs de cette espèce présentes dans ce bouquet. Par exemple, le colorama de la tulipe dans le bouquet **Belge** est le triplet contenant **noir**, **jaune**, et **rouge**. Le colorama de la tulipe dans le bouquet **Printemps** est le singleton contenant uniquement **jaune**.

Rédigez un programme en XSLT qui, pour chaque espèce de fleur, liste les noms des bouquets contenant cette espèce, ainsi que le colorama de cette espèce dans chaque bouquet. **Votre programme doit se limiter aux instructions XSLT enseignées au cours. Notamment, il n'est pas permis d'utiliser `xsl:for-each` et `xsl:if`. Il n'est pas non plus permis d'utiliser la fonction `distinct-values` ou l'instruction `<xsl:with-param>`.** Les résultats doivent être groupés comme illustrés ci-dessous, en utilisant exactement les mêmes balises et attributs :

```
<colorama>
<f flower="tulipe">
  <b bouquet="Belge"><c color="noir" /><c color="jaune" /><c color="rouge" /></b>
  <b bouquet="Printemps"><c color="jaune" /></b>
</f>
<f flower="rose">
  <b bouquet="Valentin"><c color="rouge" /><c color="blanc" /></b>
  <b bouquet="Printemps"><c color="jaune" /></b>
</f>
<f flower="iris">
  <b bouquet="Exotique"><c color="bleu" /></b>
</f>
<f flower="tournesol">
  <b bouquet="Exotique"><c color="jaune" /></b>
</f>
<f flower="dahlia">
  <b bouquet="Printemps"><c color="rose" /></b>
</f>
</colorama>
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
<colorama>
  <xsl:apply-templates select="//fleurs/fleur"/>
</colorama>
</xsl:template>

<xsl:template match="fleurs/fleur">
<f flower="{@fnom}">
  <xsl:apply-templates select="//bouquet/fleur[@fnom=current()/@fnom][1]"/>
</f>
</xsl:template>

<xsl:template match="bouquet/fleur">
<b bouquet="{parent::bouquet/@bnom}">
  <xsl:apply-templates select="parent::bouquet/fleur[@fnom=current()/@fnom]/@couleur"/>
</b>
</xsl:template>

<xsl:template match="@couleur">
  <c color="{.}"/>
</xsl:template>

</xsl:stylesheet>
```

Question 4 Rédigez une requête en **XQuery** pour résoudre le même problème que celui décrit à la question 3, en veillant à formater le résultat de manière identique.

FORMULEZ VOTRE REQUÊTE DE MANIÈRE À RENDRE SA LOGIQUE AISÉMENT COMPRÉHENSIBLE.

.../10

```
<colorama>{
for $f in //fleurs/fleur/@fnom
return <f flower="{ $f }">{
  for $b in //bouquet[fleur/@fnom=$f]
  return <b bouquet="{ $b/@bnom }">{
    for $c in $b/fleur[@fnom=$f]/@couleur
    return <c color="{ $c }"/>
  }</b>
}</f>
}</colorama>
```

Question 5 Pour un problème de classification à deux classes (par exemple, “yes” et “no”), l’exactitude (connue en anglais sous le nom d’*accuracy*) d’un classificateur est définie comme $ACCURACY = \frac{TP+TN}{P+N}$. Deux autres métriques abordées en cours sont le TPR (taux de vrais positifs) et le FPR (taux de faux positifs). Le TPR est à maximiser, tandis que le FPR est à minimiser. De plus, si $TPR \leq FPR$ pour un classificateur, alors sa performance n’est pas meilleure que celle d’un classificateur aléatoire (et peut même être pire). Cela soulève une question pertinente :

Pour un problème de classification à deux classes (“yes” et “no”), existe-t-il des classificateurs avec une exactitude élevée (par exemple, $ACCURACY \geq 0.8$), mais pour lesquels $TPR \leq FPR$?

.../10

Complétez les cases vides de manière à obtenir une matrice de confusion où la valeur d’ACCURACY dépasse 0.8, mais où le classificateur sous-jacent n’est pas meilleur qu’un classificateur aléatoire

		Classe prédite	
		yes	no
Classe observée	yes	10	90
	no	90	810

Dans l’espace ci-dessous, détaillez également un scénario concret tiré de la vie quotidienne qui correspond à une telle matrice de confusion. Par exemple, un exemple basé sur la détection de spam, la classification de maladies, ou tout autre contexte approprié.

On obtient $TPR = \frac{10}{100}$ et $FPR = \frac{90}{900}$, donc $TPR = FPR = 0.1$. L’exactitude est de $\frac{820}{1000} > 0.8$.

Cela correspond à un classificateur qui prédit, de manière aléatoire et sans examiner les données, “oui” avec une probabilité de $\frac{1}{10}$ (et donc “non” avec une probabilité de $\frac{9}{10}$). La probabilité qu’une instance “oui” soit correctement prédite comme “oui” est donc de $\frac{1}{10}$ (= TPR). La probabilité qu’une instance “non” soit incorrectement prédite comme “oui” est également de $\frac{1}{10}$ (= FPR).

L’exactitude est élevée en raison du nombre important de TN (le nombre de 810).

Scénario concret. Le cirque Bouglione souhaite faire connaître son futur spectacle à travers des dépliants publicitaires envoyés à des familles dans la région bruxelloise. À cette fin, le chef marketing s’est procuré une base de données contenant des informations sur 1000 familles, telles que l’âge et le nombre d’enfants, l’adresse postale, etc. Il sait que, dans une telle campagne publicitaire, en général, 10% des destinataires répondent favorablement au dépliant reçu. En d’autres termes, envoyer le dépliant à toutes les adresses de la base de données aboutirait à 100 familles visitant le spectacle. Cependant, en raison du coût élevé de l’impression et de l’envoi des dépliants, seuls 100 dépliants peuvent être envoyés.

La question est donc d’identifier, parmi les 1000 familles de la base de données, celles qui sont les plus susceptibles de répondre favorablement à un dépliant. Il s’agit donc de classer les familles en deux classes : susceptibles ou non susceptibles de répondre favorablement. Les dépliants seront ensuite envoyés uniquement aux familles classées comme susceptibles. Dans ce contexte, un vrai positif (true positive) désigne une famille qui reçoit un dépliant et assiste au spectacle, tandis qu’un faux positif désigne une famille qui reçoit un dépliant mais n’assiste pas au spectacle.

Le classificateur aléatoire décrit plus haut aboutit à une matrice de confusion similaire à celle dessinée ci-dessus. L’exactitude est élevée en raison des 810 familles qui ne reçoivent pas de dépliant et n’assistent pas au spectacle. Cependant, ce classificateur est loin d’être idéal, car 90 familles reçoivent un dépliant sans y répondre favorablement. L’objectif est d’obtenir un classificateur avec un TPR proche de 1, c’est-à-dire que presque tous les dépliants envoyés entraînent une visite.

Question 6 Un problème de classification important consiste à classer les emails en deux catégories : Spam et Non-spam. Supposons que nous disposons des probabilités prédites par Naive Bayes pour 10 emails, ainsi que de la vérité terrain (indiquant s'ils sont réellement des spams ou non). Les données sont présentées dans le tableau suivant :

Email	Probabilité de spam	Vérité terrain
1	0.95	Spam
2	0.85	Non-spam
3	0.80	Spam
4	0.75	Non-spam
5	0.70	Spam
6	0.65	Spam
7	0.60	Non-spam
8	0.55	Non-spam
9	0.50	Spam
10	0.45	Non-spam

Nous souhaitons définir une règle de décision simple pour classer les nouveaux emails, de la forme suivante :

Si la *Probabilité de spam* est *strictement* supérieure à un seuil x , alors l'email est classé comme Spam ;
 sinon, l'email est classé comme Non-spam.

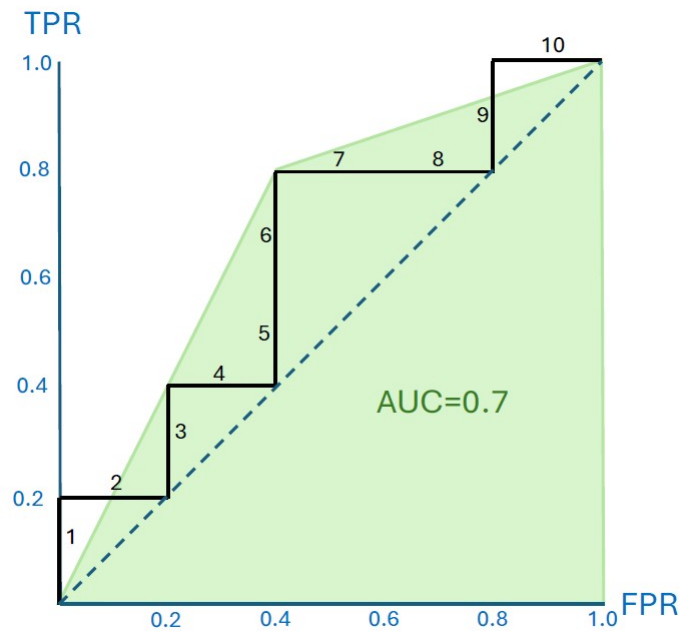
L'objectif est de maximiser l'AUC (Area Under the Curve) dans un plan où l'abscisse représente le FPR (False Positive Rate) et l'ordonnée représente le TPR (True Positive Rate). Indiquez d'abord, dans les cases prévues, la valeur du seuil x choisi ainsi que la valeur de l'AUC correspondante. Détaillez ensuite les calculs qui vous ont permis de déterminer ce seuil optimal x et la valeur de l'AUC correspondante.

.../10

La valeur de x est : .

La valeur de l'AUC est : .

Détaillez ci-dessous les calculs qui vous ont permis de déterminer ce seuil optimal x et la valeur AUC correspondante.



Le graphique permet de déterminer facilement les valeurs de FPR et TPR obtenues pour un seuil x choisi entre les probabilités des emails i et $i + 1$ (pour tout $i \in \{1, 2, \dots, 9\}$).

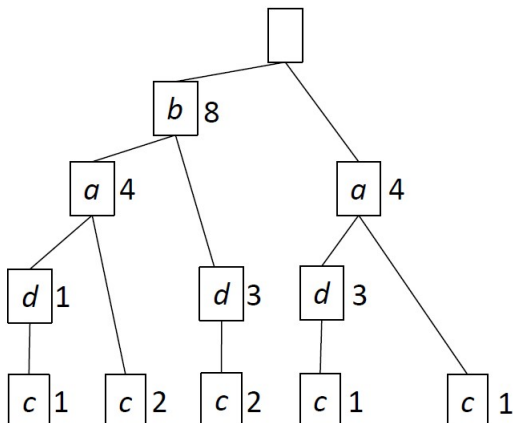
Par exemple, si on choisit $x = 0.77$ (c'est-à-dire entre les probabilités des emails 3 et 4), on obtient FPR=0.2 et TPR=0.4. En effet, en appliquant la règle "*si $x \geq 0.77$, alors Spam; sinon, Non-spam*", on identifie 2 vrais positifs (les emails 1 et 3) et un faux positif (l'email 2). Puisque P=N=5, on obtient TPR= $\frac{2}{5} = 0.4$ et FPR= $\frac{1}{5} = 0.2$.

Le graphique permet également de constater que l'AUC (la surface verte) est maximale lorsque x est choisi entre les probabilités des emails 6 et 7. La surface verte au-dessus de la diagonal est de 0.2. Cette surface correspond à un triangle dont la base est de longueur $\sqrt{2}$ (la distance entre les points (0,0) et (1,1)) et la hauteur est $\sqrt{2}/5$ (la distance entre les points (0.4,0.8) et (0.6,0.6)). Ainsi, l'AUC pour $x = 0.62$ est $0.5 + 0.2 = 0.7$.

Question 7 Voici l'arbre FP (connu en anglais sous le nom de *FP-tree*) associé à une base de données contenant des transactions composées des articles a, b, c, d . L'ordre lexicographique utilisé pour construire cet arbre est le suivant :

$$b < a < d < c.$$

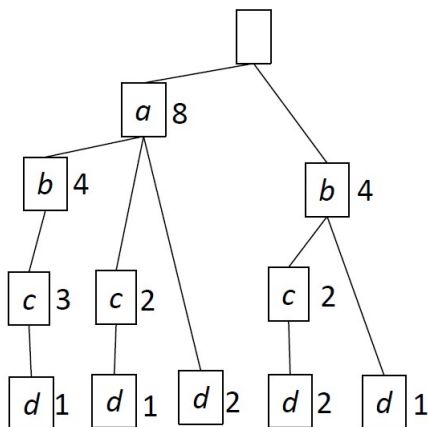
Donc, on a utilisé un ordre où b précède a et d précède c .



Pour la même base de données, dessinez d'abord l'arbre FP obtenu si l'on utilisait l'ordre lexicographique standard $a < b < c < d$. Enfin, détaillez les étapes qui vous ont permis de construire votre arbre FP.

.../10

Dessinez ci-dessous l'arbre FP construit en utilisant l'ordre lexicographique standard $a < b < c < d$:



Détaillez ci-dessous les étapes qui vous ont permis de construire votre arbre FP.

L'arbre FP original encode l'ensemble suivant de transactions :

badc
bac
bac
ba
bdc
bdc
bd
b
adc
ad
ad
ac

En ordre alphabétique :

abcd
abc
abc
ab
bcd
bcd
bd
b
acd
ad
ad
ac

Créer l'arbre FP pour cet ensemble.

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE fleuriste [
<!ELEMENT fleuriste (fleurs, bouquets)>
<!ELEMENT fleurs (fleur)*>
<!ELEMENT bouquets (bouquet)*>
<!ELEMENT bouquet (fleur)*>
<!ELEMENT fleur (#PCDATA)>
<!ATTLIST fleur fnom CDATA #REQUIRED>
<!ATTLIST fleur prix CDATA #IMPLIED>
<!ATTLIST fleur couleur CDATA #IMPLIED>
<!ATTLIST fleur nombre CDATA #IMPLIED>
<!ATTLIST bouquet bnom CDATA #REQUIRED>
]>

<fleuriste>
  <fleurs>
    <!-- Les prix sont en centimes d'euro -->
    <fleur fnom="tulipe" prix="100"/>
    <fleur fnom="rose" prix="150"/>
    <fleur fnom="iris" prix="250"/>
    <fleur fnom="tournesol" prix="300"/>
    <fleur fnom="dahlia" prix="120"/>
  </fleurs>
  <bouquets>
    <bouquet bnom="Valentin">
      <fleur fnom="rose" couleur="rouge" nombre="10"/>
      <fleur fnom="rose" couleur="blanc" nombre="10"/>
    </bouquet>
    <bouquet bnom="Belge">
      <fleur fnom="tulipe" couleur="noir" nombre="3"/>
      <fleur fnom="tulipe" couleur="jaune" nombre="4"/>
      <fleur fnom="tulipe" couleur="rouge" nombre="6"/>
    </bouquet>
    <bouquet bnom="Exotique">
      <fleur fnom="tournesol" couleur="jaune" nombre="1"/>
      <fleur fnom="iris" couleur="bleu" nombre="3"/>
    </bouquet>
    <bouquet bnom="Printemps">
      <fleur fnom="rose" couleur="jaune" nombre="10"/>
      <fleur fnom="tulipe" couleur="jaune" nombre="4"/>
      <fleur fnom="dahlia" couleur="rose" nombre="3"/>
    </bouquet>
  </bouquets>
</fleuriste>

```

FIGURE 1 – Document XML précédé de son DTD

Feuille de brouillon.