

CT JS/PHP : Session 2 - 1h30 - Sur machine

Téléchargez et décompressez l'archive déposée sur Moodle pour cet examen. Le dossier résultant contient différents fichiers à réutiliser ou à compléter. Vous testerez vos scripts en ligne de commande ou avec le conteneur Docker et Firefox selon les exercices. A l'issue du temps imparti, archivez votre dossier et déposez l'archive sur Moodle.

Le sujet comporte 3 exercices indépendants portant sur les cartes cognitives. Une carte cognitive est un graphe orienté (digraphe) dont les nœuds sont étiquetés par des concepts et les arcs par des valeurs qualifiant "l'influence" entre concepts. Le domaine d'influences est laissé au choix. Par exemple, la carte illustrée en Figure 1 est étiquetée sur $\{-, +\}$.

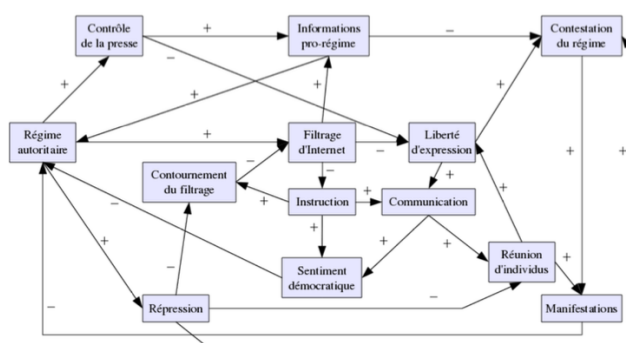


FIGURE 1 – Carte étiquetée sur $\{-, +\}$

Exercice 1. [Javascript]

On propose une page web pour visualiser les cartes cognitives stockées dans les fichiers JSON du dossier **data**. Toutes sont étiquetées avec les symboles $+$ et $-$ et suivent le même schéma de représentation.

La page comprend un formulaire permettant de sélectionner une carte par menu déroulant. Le fichier JSON correspondant est alors requêté et sa carte affichée sous forme matricielle. Notez que la carte sélectionnée par défaut est codée en dur et s'affiche sans requêtage au chargement de la page (voir Figure 2). Le formulaire fournit deux boutons radio permettant de colorier alternativement arcs positifs et arcs négatifs (voir Figure 3). Il permet aussi de déterminer les chemins simples reliant un noeud origine à un noeud destination. Le calcul est délégué à un script PHP et les chemins sont affichés en console (voir Figure 4). Consultez ce [démonstrateur](#) pour tester les différentes fonctionnalités.

La page est fournie par le fichier **map.html** qui importe le fichier obfusqué **matrice.js** et le fichier **map.js** que vous complétez. Ce dernier initialise la variable objet `map_3_3_acyclic` avec une chaîne au format JSON identique à la carte du fichier **map_3_3_acyclic.json**. Cette carte est affichée au chargement par l'appel `matrice(map_3_3_acyclic)`. La fonction `matrice` est implémentée dans **matrice.js** et documentée dans **map.js**. Elle fait elle-même appel à la fonction `influence` pour afficher les coefficients, fonction que vous développerez dans l'exercice 1.

1. La fonction `influence(...)` est utilisée pour extraire une étiquette d'arc. Implémentez-la en suivant les instructions données en commentaires dans le code source.
2. La fonction `champ(...)` est utilisée pour transformer les champs du formulaire selon la carte sélectionnée. Implémentez-la en suivant les instructions données en commentaires dans le code source.
3. Implémentez un écouteur sur les boutons radio :
 - tout clic sur le bouton "POSITIF" place les cellules de coefficient $+$ dans la classe HTML `positif` et retire les cellules de coefficient $-$ de la classe `negatif`
 - tout clic sur le bouton "NEGATIF" place les cellules de coefficient $-$ dans la classe HTML `negatif` et retire les cellules de coefficient $+$ de la classe `positif`.

CARTE

acyclic

noeuds

3

arcs

3

type

symbolic

POSITIF

☐

NÉGATIF

☐

ORIGINE

1

DESTINATION

3

chemins

	e1	e2	e3
e1		+	+
e2			-
e3			

FIGURE 2 – Page d'accueil au chargement

CARTE

acyclic

noeuds

3

arcs

3

type

symbolic

POSITIF

☒

NÉGATIF

☐

ORIGINE

1

DESTINATION

3

chemins

	e1	e2	e3
e1		+	+
e2			-
e3			

FIGURE 3 – Coloration des étiquettes +

CARTE

acyclic

noeuds

3

arcs

3

type

symbolic

POSITIF

☐

NÉGATIF

☒

ORIGINE

1

DESTINATION

3

chemins

	e1	e2	e3
e1		+	+
e2			-
e3			

Inspecteur

Console

Réseau

Débogueur

Erreurs

Avertissements

Journaux

Informations

Débugage

CSS

XHR

Requêtes

POST http://localhost:8080/l3dw/ct-23-2/demonstrateur- [HTTP/1.1 200 OK 5ms]

console.table()

(index)

Valeurs

0

1

1

2

2

3

console.table()

(index)

Valeurs

0

1

1

3

FIGURE 4 – Calcul de chemins

4. La sélection d'une carte à l'aide du menu déroulant déclenche une requête Ajax qui récupère le fichier JSON correspondant par méthode HTTP GET :

- le nom du fichier correspond à l'attribut value de l'option choisie, suffixé par .json
- le fichier se situe dans le dossier **/data**.

Formulaire et tableau HTML sont ensuite réactualisés par appel aux fonctions champs et matrice. Implémentez cette fonctionnalité en utilisant l'API Fetch.

5. Un clic sur le bouton "chemins" affiche les chemins simples reliant les noeuds origine et destination renseignés dans les champs numériques. Le calcul est délégué au script **/php/api.php** auquel les champs suivants sont communiqués par requête HTTP POST :

- carte : le nom de la carte
- origine : le numéro du noeud origine (≥ 1)
- destination : le numéro du noeud destination (≥ 1).

Les chemins sont renvoyés au format JSON sous forme de tableau de tableaux. Par exemple, la requête paramétrée par `carte=map_3_3_acyclic&origine=1&destination=3` renvoie le tableau `[[1, 2, 3], [1, 3]]` des 2 chemins possibles. Les chemins récupérés et les messages d'erreurs sont simplement affichés en console. Implémentez cette fonctionnalité en utilisant l'API Fetch.

Exercice 2. [PHP : Base de données et génération de HTML]

On souhaite afficher les matrices d'adjacence de cartes cognitives stockées dans une base de données MySQL. L'objectif est d'implémenter les fonctions nécessaires pour récupérer et afficher ces cartes.

Importez d'abord le script `./utils/map.sql` pour créer la base de données `l3_ct_23_map` dont le schéma est donné en Figure 5 (attention : tout est compris dans le script, vous n'avez pas besoin de créer la base de données à l'avance). Complétez les fonctions du fichier `bdd_cartes.php` et le script `exercice2.php` pour répondre aux questions qui suivent en utilisant l'API PDO quand nécessaire.

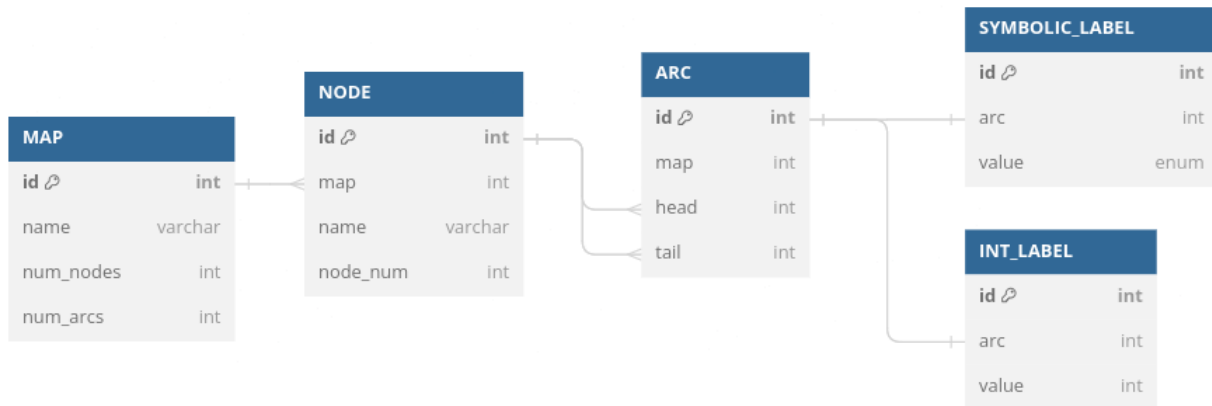


FIGURE 5 – Schéma de la base de données `l3_ct_23_map`

1. Implémentez la fonction `bdd_get_map(int $map_id)` pour récupérer les paramètres d'une carte cognitive (nom de la carte, nombre de nœuds et nombre d'arcs). La fonction retournera le tableau associatif des paramètres si la carte existe, ou `null` en cas d'échec.
2. Implémentez la fonction `bdd_get_arcs(int $map_id)` pour récupérer l'ensemble d'arcs d'une carte cognitive. La fonction retournera un tableau associatif (l'identifiant d'un arc servant de clé) si la carte existe, ou `null` en cas d'échec. La requête SQL à utiliser est fournie dans le fichier `bdd_cartes.php`.
3. Implémentez la fonction `get_arc_label(int $x, int $y, array $arcs)` pour récupérer l'étiquette d'un arc. La fonction prend en paramètres deux identifiants de nœuds et un tableau d'arcs du type de ceux retournés par la fonction `bdd_get_arcs(int $map_id)` (voir question précédente). Retournez `null` si l'arc (x, y) ne figure pas dans le tableau ou s'il est sans étiquette. Si vous n'avez pas implémenté les fonctions précédentes, utilisez les tableaux fournis dans `./utils/data.php`.
4. Implémentez la fonction `afficher_matrice_adj(int $nb_noeuds, array $arcs)` qui prend en paramètres un nombre de nœuds et un tableau d'arcs du type de ceux retournés par la fonction `bdd_get_arcs(int $map_id)`. La fonction doit afficher la matrice d'adjacence de la carte qui correspond (voir Figure 6).
5. Dans le fichier `exercice2.php`, implémentez la partie manquante en appelant les fonctions précédentes pour récupérer et afficher une carte cognitive.

Service d'affichage de cartes cognitives

Choisissez une carte cognitive:

Map_1 ▼ Afficher

	n1	n2	n3
n1			
n2	+		
n3		+	

FIGURE 6 – Affichage d'une carte cognitive

Exercice 3. [PHP : Sessions/cookies]

1. Le service d'affichage de cartes ne peut fonctionner que 15 minutes d'affilée : lorsque le visiteur attend 15 minutes pour rafraîchir la page, le serveur le redirige vers une autre page. Plus précisément, les comportements suivants sont attendus (lors du rafraîchissement de page) :

- Après 1 minute, la couleur de fond de la barre en haut (`id=datetime-bar`) change (la couleur par défaut est bleu). Vous pouvez utiliser la fonction `random_color()` fournie dans la bibliothèque `./utils/colors.php`.
- Après 10 minutes, le service d'affichage lance une alerte avec le message : *"Attention! Le temps s'écoule!"* et change la couleur de fond de la barre du haut. Vous pouvez utiliser la fonction `alert()` fournie dans `./utils/js.php`.
- Après 15 minutes, le serveur redirige vers le site web du jeu **Time's Up!**. Un exemple codé pour rediriger vers une autre page vous est fourni.

Implémentez ce comportement dans le fichier `exercice3.php` en utilisant des variables de session.