

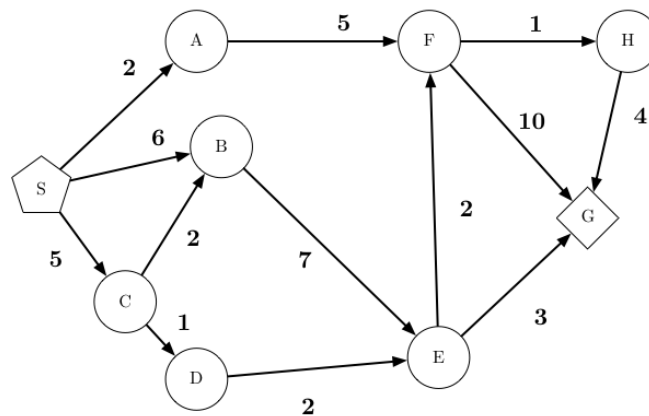
INF8215 : Examen final (Automne 2021) - Quentin Cappart			Note finale ↓
Date : 22/12/2021	Nombre de questions : 6	Total des points : 30	
Matricule :	Nom :	Temps : 2h30 heures	

Question:	1	2	3	4	5	6	Total
Points:	5	5	5	5	5	5	30
Score:							

Instructions

1. La documentation papier est autorisée.
2. Les dispositifs électroniques ne sont pas autorisés.
3. L'examen doit se réaliser de manière individuelle. Il est strictement interdit de collaborer.

- 5 1. Soit le graphe des états présenté ci-dessous. Chaque nœud correspond à un état spécifique, chaque arête à une action possible ayant un coût spécifique. L'état S est l'état initial, et l'état G est l'état final. L'objectif est de déterminer une séquence d'action vous amenant de S à G. Pour toutes les sous-questions, on va considérer *une recherche en graphe* (et non en arbre). Précisément, un état ne sera pas ajouté dans la liste des états candidats (*fringe*) si et seulement s'il a déjà été visité. La séquence optimale est celle minimisant la somme des coûts des actions effectuées.



- Déterminez la séquence d'actions qui sera obtenue en exécutant *une recherche en largeur*. Indiquez chaque étape de l'algorithme. À chaque étape, veillez à bien indiquer votre état actuel, ainsi que les nœuds présents dans la liste des états candidats (*fringe*). Dans le cas d'une priorité égale lors de l'ajout d'un successeur, ajoutez d'abord le successeur selon un critère alphabétique (la première lettre d'abord).
- Déterminez la séquence d'actions qui sera obtenue en exécutant *une recherche A**. Considérez comme fonction heuristique le nombre d'actions possibles (le nombre d'arêtes sortantes) au nœud multiplié par 5. À titre d'exemple, l'heuristique appliquée au nœud S vaut 15 (3×5). Indiquez chaque étape de l'algorithme. À chaque étape, veillez à bien indiquer votre état actuel, ainsi que les nœuds présents dans la liste des états candidats (*fringe*). Dans le cas d'une priorité égale lors de l'ajout d'un successeur, ajoutez d'abord le successeur selon un critère alphabétique (la première lettre d'abord).
- Pour ces deux algorithmes et la fonction heuristique choisie, et supposant un temps d'exécution infini, avez-vous la garantie que votre algorithme trouve une solution ? Justifiez votre réponse.
- Pour ces deux algorithmes et la fonction heuristique choisie, et supposant un temps d'exécution infini, avez-vous la garantie que votre algorithme trouve la solution optimale ? Justifiez votre réponse.

Solution:**1. L'état le plus vieux dans la liste est étendu en priorité**

Etape 1 [state : S , fringe : $\langle A, B, C \rangle$]

Etape 2 [state : A , fringe : $\langle B, C, F \rangle$]

Etape 3 [state : B , fringe : $\langle C, F, E \rangle$]

Etape 4 [state : C , fringe : $\langle F, E, D \rangle$] (B non ajouté car déjà visité)

Etape 5 [state : F , fringe : $\langle E, D, G^F, H \rangle$]

Etape 6 [state : E , fringe : $\langle D, G^F, H, G^E \rangle$]

Etape 7 [state : D , fringe : $\langle G^F, H, G^E \rangle$] (E non ajouté car déjà visité)

Etape 8 [state : G^F] (état final atteint)

Solution $S \rightarrow A \rightarrow F \rightarrow G$ (coût de 17)

2. L'état ayant le plus petit score heuristique + coût est étendu en priorité

Etape 1 [state : S , fringe : $\langle (A, 7)^S, (B, 11)^S, (C, 15)^S \rangle$]

Etape 2 [state : A , fringe : $\langle (B, 11)^S, (C, 15)^S, (F, 17)^A \rangle$]

Etape 3 [state : B , fringe : $\langle (C, 15)^S, (F, 17)^A, (E, 23)^B \rangle$]

Etape 4 [state : C , fringe : $\langle (F, 17)^A, (E, 23)^B, (D, 11)^C \rangle$] (B non ajouté car déjà visité)

Etape 5 [state : D , fringe : $\langle (F, 17)^A, (E, 23)^B, (E, 18)^D \rangle$]

Etape 6 [state : F , fringe : $\langle (E, 23)^B, (E, 18)^D, (G, 17)^F, (H, 13)^F \rangle$]

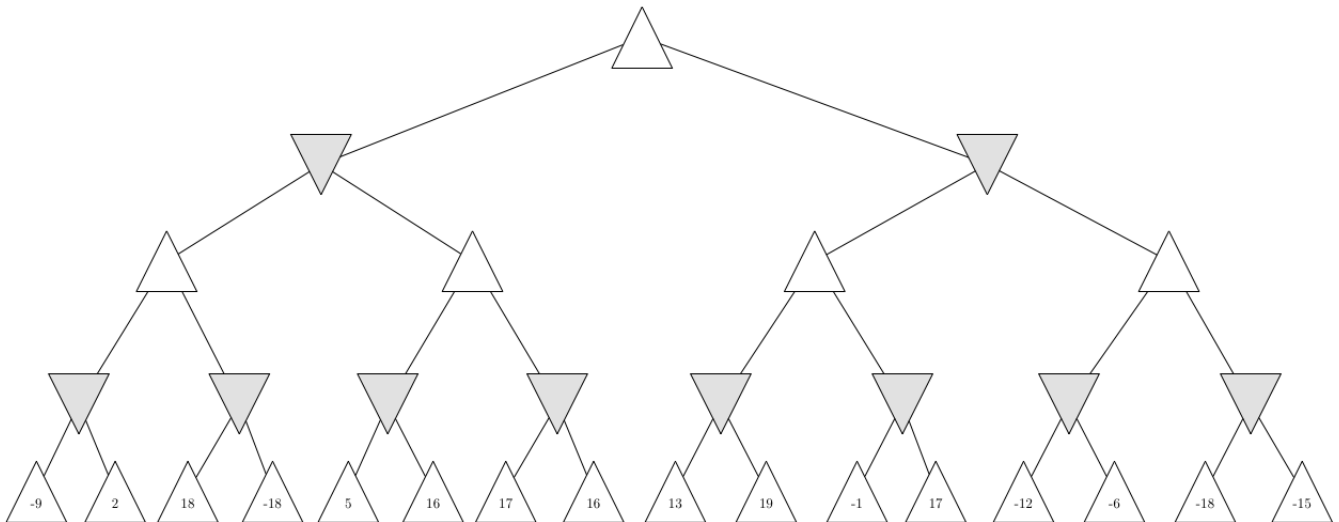
Etape 7 [state : H , fringe : $\langle (E, 23)^B, (E, 18)^D, (G, 17)^F, (G, 12)^H \rangle$]

Etape 8 [state : G^H] (état final atteint)

Solution $S \rightarrow A \rightarrow F \rightarrow H \rightarrow G$ (coût de 12)

3. Les deux algorithmes sont complets, on progresse niveau par niveau soit suivant le nombre d'actions (recherche en profondeur), soit en utilisant le coût déjà engendré (A^*). Dans le cas de A^* , c'est bon car tous les coûts sont positifs et la solution a un coût fini.
4. La recherche en largeur n'est pas optimale, car la meilleure solution n'est pas celle impliquant le moins d'actions. La recherche A^* n'est pas optimale car l'heuristique n'est pas admissible. Par exemple, l'heuristique appliquée en H (5) surestime le coût pour atteindre G (4).

- 5 2. Considérons un jeu déterministe, à deux joueurs (**MAX** et **MIN**), tour par tour, à information parfaite, et à somme nulle. L'arbre de recherche avec le score atteint par le joueur **MAX** est présenté ci-dessous.

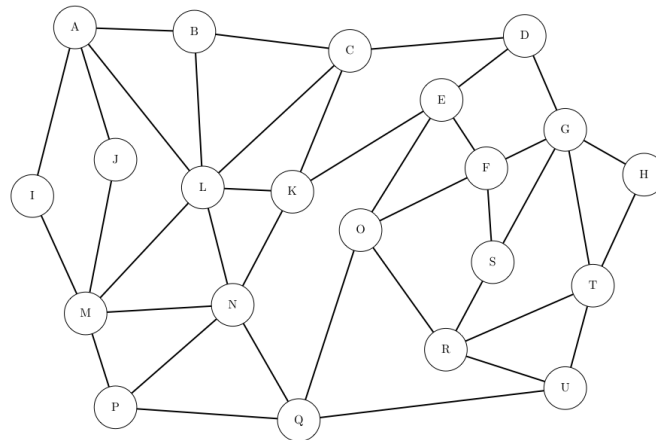


1. Remplissez la valeur minimax de tous les nœuds non-terminaux. Il n'est pas nécessaire de justifier votre raisonnement. Une seule valeur non correcte à cette question vous fait perdre la totalité de points pour cette sous-question.
2. On utilise maintenant l'algorithme *alpha-beta pruning* pour réduire l'espace de recherche. Indiquez quelles branches seront supprimées lors de l'application de l'algorithme. Considérez que l'exploration se fait toujours sur la branche de gauche en priorité (même convention que celle utilisée dans les slides). Il n'est pas nécessaire de justifier votre raisonnement. Une seule valeur non correcte à cette question vous fait perdre la totalité de points pour cette sous-question.
3. Supposons que les deux joueurs sont rationnels, quel sera le score obtenu par le joueur **MAX** ?
4. Supposons maintenant que vous êtes un joueur rationnel suivant une stratégie minimax (**MAX**) et que le joueur adverse (**MIN**) est non-rationnel (comportement non prévisible). Quel est le score minimal que vous êtes assuré d'avoir ? Quel est le score maximal que vous pouvez potentiellement atteindre ?

The diagram illustrates a minimax search tree for a game. The root node is a triangle labeled -9, with children triangles labeled -9 and -12. The -9 node has children triangles labeled -9 and 5. The -12 node has children triangles labeled 13 and -12. The 5 node has children triangles labeled 5 and a dashed triangle. The 13 node has children triangles labeled 13 and -1. The -12 node has children triangles labeled -12 and -18. The leaf nodes are squares: -9, 2, 18, -18, 5, 16, 17, 16, 13, 19, -1, 17, -12, -6, -18, -15. The diagram shows the progression of the minimax algorithm, with nodes and edges highlighted in red to indicate the current state of the search.

- Page 4 / 12

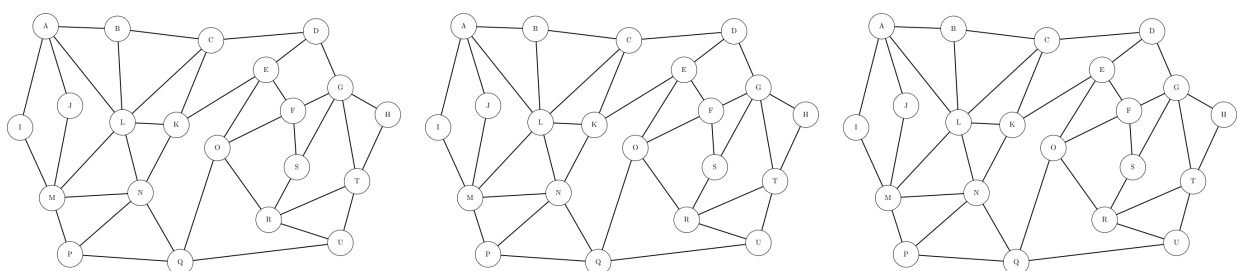
- 5 3. Pour le temps des fêtes, vous souhaitez organiser une soirée entre amis chez vous. Vos amis ont des relations assez difficiles entre eux, et certains sont en conflit. Chaque conflit est réciproque, c'est-à-dire que si Alice est en conflit avec Bernard, Bernard sera également en conflit avec Alice. L'ensemble des conflits est représenté comme un graphe, où chaque nœud correspond à un ami, et chaque arête indique la présence d'un conflit. Pour vos 21 amis, le graphe des conflits est présenté ci-dessous.



Afin que tout le monde profite au mieux de la soirée, vous décidez de n'inviter que des amis qui ne sont pas en conflit. Malheureusement, suite aux nouvelles mesures sanitaires, vous êtes également limité à n'inviter que 10 amis. Votre problème est le suivant : *"Est-il possible d'inviter exactement 10 amis de sorte à ce qu'il n'y ait aucun conflit ?"*. Par exemple, il n'est pas possible d'inviter à la fois Alice (A) et Bernard (B), par contre il est possible d'inviter Alice (A) et Claude (C).

Pour cette question, il ne vous est pas demandé de résoudre complètement ce problème, mais simplement de proposer une résolution basée sur la recherche locale et d'exécuter quelques étapes de l'algorithme.

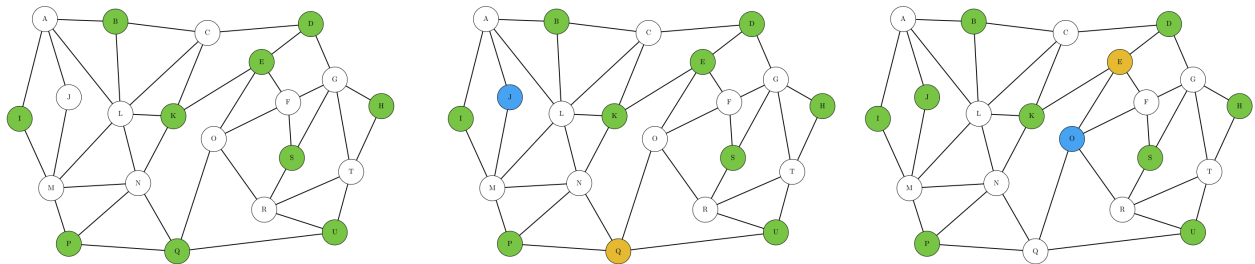
- Proposez un modèle de recherche locale en vue de résoudre ce problème. En particulier, veillez à bien définir les éléments suivants :
 - Votre espace de recherche. Indiquez le plus précisément possible quelle est sa taille.
 - Les contraintes dures et molles relatives à l'espace de recherche que vous avez considéré.
 - Une méthode pour obtenir une solution initiale.
 - Votre fonction d'évaluation pour caractériser la qualité d'un état.
 - Votre fonction de voisinage.
 - Votre critère pour sélectionner un voisin dans le voisinage.
- Exécutez deux itérations d'une recherche locale sur base de votre formulation. Vous ne devez pas indiquer tous les voisins de votre voisinage, prenez en juste 5 au maximum (qui sont cohérents avec la définition de votre voisinage), et faites la sélection parmi ces 5 voisins selon votre critère. Dans les figures ci-dessous, illustrez (a) votre solution initiale, (b) la solution obtenue après la première itération, (c) la solution obtenue après la deuxième itération. Pour ces trois solutions, indiquez la valeur de votre fonction d'évaluation.



- Proposez une méthode pour éviter que votre algorithme se retrouve bloqué dans un minimum local.

Solution:

1. Il existe plusieurs modélisations possibles. Le plus important est que votre modèle soit cohérent. Par exemple, votre voisinage ne doit considérer que des solutions qui sont conformes par rapport à vos contraintes dures. Un modèle facile à mettre en place est le suivant.
 - (a) Pour chaque ami, on a une variable binaire indiquant s'il est invité ou non. On force également à avoir exactement 10 amis invités. Sachant qu'on a 21 amis dans notre problème, on a ainsi un espace de recherche de $21 \times 20 \times \dots (21 - 10)$ (21 possibilités pour le premier ami invité, 20 pour le second, et ainsi de suite jusqu'au dixième).
 - (b) La contrainte dure est que exactement 10 amis sont invités. La contrainte molle est que les conflits peuvent être non-satisfaits dans notre espace de recherche.
 - (c) On peut simplement inviter 10 amis au hasard (voir la figure de gauche ci-dessous).
 - (d) On peut compter le nombre de conflits de la solution actuelle. Pour notre situation initiale, on a 4 conflits (P et Q , Q et U , K et E , D et E).
 - (e) On peut considérer toutes les permutations possibles entre un ami invité et non invité.
 - (f) On peut sélectionner un voisin aléatoire du voisinage qui réduit le coût. Dans les deux itérations, on permute Q avec J (nouveau coût de 2), et ensuite O avec E (nouveau coût de 0). Dans ce cas particulier, on a trouvé une solution faisable.
2. La solution initiale et les deux itérations sont présentées ci-dessous. Notez qu'en général, on a pas la garantie de converger vers le minimum global.
3. On peut utiliser des restarts aléatoires. On commence à partir d'une nouvelle situation initiale, et on répète un certain nombre de fois.



- 5 4. Soit le problème de satisfaction combinatoire ci-dessous.

satisfy

$$\text{subject to } x = y^2 \quad (c_1)$$

$$z = 3y + 2 \quad (c_2)$$

$$x, y, z \in \{0, \dots, 10\}$$

1. Indiquez la taille de l'espace de recherche si on souhaite réaliser une recherche exhaustive.
2. Appliquez l'algorithme du point fixe jusqu'à convergence afin de réduire l'espace de recherche. Supposez que le propagateur de chaque contrainte vérifie la cohérence d'arc.
3. Indiquez la taille restante de l'espace de recherche.

Solution:

1. Le produit cartésien des domaines : $11 \times 11 \times 11 = 1331$ solutions possibles.
2. A l'initialisation, les deux contraintes (c_1 et c_2) sont ajoutés dans la queue de propagation (Q).

Initialisation [$Q = \langle c_1, c_2 \rangle, x \in \{0, \dots, 10\}, y \in \{0, \dots, 10\}, z \in \{0, \dots, 10\}$]

Prop. de c_1 [$Q = \langle c_2 \rangle, x \in \{0, 1, 4, 9\}, y \in \{0, 1, 2, 3\}, z \in \{0, \dots, 10\}$]

Prop. de c_2 [$Q = \langle c_1 \rangle, x \in \{0, 1, 4, 9\}, y \in \{0, 1, 2\}, z \in \{2, 5, 8\}$] (ajout de c_1 car y a changé)

Prop. de c_1 [$Q = \langle \rangle, x \in \{0, 1, 4\}, y \in \{0, 1, 2\}, z \in \{2, 5, 8\}$]

Convergence La queue de propagation est vide.

3. Le produit cartésien des domaines : $3 \times 3 \times 3 = 27$ solutions possibles.

5. Pour l'entièreté des questions, on se situe dans la logique des propositions, sauf si le contraire est explicitement mentionné. Indiquez si les assertions suivantes sont vraies ou fausses. Si elles sont fausses, justifiez succinctement (deux phrases au maximum) pourquoi ou donnez un contre-exemple.
1. Soit la base de connaissances KB : $\{\text{Toux}, \text{Fievre} \wedge \text{Toux} \rightarrow \text{Grippe}\}$. La formule $f_1 : \neg(\text{Fievre} \vee \text{Toux})$ est en contradiction avec KB.
 2. Soit la base de connaissances KB : $\{\text{Toux}, \text{Fievre} \wedge \text{Toux} \rightarrow \text{Grippe}\}$. La formule $f_2 : \text{Grippe}$ est une conséquence logique de KB.
 3. Soit la base de connaissances KB : $\{\text{Toux}, \text{Fievre} \wedge \text{Toux} \rightarrow \text{Grippe}\}$. La formule $f_3 : \neg \text{Fievre}$ est en contingence avec KB.
 4. Soit la base de connaissances KB : $\{\text{Toux}, \text{Fievre} \wedge \text{Toux} \rightarrow \text{Grippe}\}$. Cette dernière est non-satisfiable.
 5. La règle de résolution est cohérente (*sound*) et complète dans la logique des propositions limitée aux clauses de Horn.
 6. La règle du Modus Ponens est cohérente dans la logique des propositions.
 7. La formule $f_4 = (P \wedge Q \wedge Z) \vee (\neg P \wedge Q) \vee (Z \wedge \neg P)$ est exprimée sous forme normale conjonctive.
 8. Il est possible qu'une phrase en français possède plusieurs sémantiques différentes.
 9. L'algorithme de résolution converge toujours dans la logique des propositions.
 10. Il est possible de définir des règles d'inférences qui ne sont pas cohérentes.

Solution:

1. Vrai : l'intersection entre les deux modèles est vide (voir figure ci-dessous).
2. Faux : l'ensemble des modèles de f_1 n'englobe pas tous les modèles de KB (voir figure ci-dessous). La formule est en contingence.
3. Vrai : l'ensemble des deux modèles se chevauchent (voir figure ci-dessous).
4. Faux : il existe au moins un modèle ($\text{Toux} = 1, \text{Fièvre} = 1, \text{Grippe} = 1$)
5. Vrai : Elle l'est dans la logique des propositions et donc dans celle limitée aux clauses de Horn.
6. Vrai : Elle ne génère que des formules en conséquences logiques. Cependant, elle n'est pas complète.
7. Faux : les \vee et les \wedge devraient être inversés pour cela.
8. Vrai : la sémantique est une notion différente de la syntaxe. On peut prendre par exemple la différence de sens entre un sens français et québécois.
9. Vrai : Il est basé sur la règle de résolution qui est complète.
10. Vrai : on peut créer n'importe quelle règle (application mécanique d'une formule). Cela ne veut pas dire que ces règles sont utiles.

KB		f1		f2		f3	
Grippe		Grippe		Grippe		Grippe	
0 1		0 1		0 1		0 1	
Fièvre, Toux	0,0						
	1,0						
	0,1						
	1,1						

KB		f1		f2		f3	
Grippe		Grippe		Grippe		Grippe	
0 1		0 1		0 1		0 1	
Fièvre, Toux	0,0						
	1,0						
	0,1						
	1,1						

KB		f1		f2		f3	
Grippe		Grippe		Grippe		Grippe	
0 1		0 1		0 1		0 1	
Fièvre, Toux	0,0						
	1,0						
	0,1						
	1,1						

KB		f1		f2		f3	
Grippe		Grippe		Grippe		Grippe	
0 1		0 1		0 1		0 1	
Fièvre, Toux	0,0						
	1,0						
	0,1						
	1,1						

- 5 6. Une compagnie de courtage immobilier décide de mettre un outil de prédiction de la valeur d'un bien. Leur objectif est d'obtenir une estimation sur le prix du marché d'un bien précis. On suppose que cette estimation prend une valeur réelle. Un bien immobilier est représenté par les 3 *features* suivantes : sa superficie, le nombre de pièces, et la présence d'un jardin ou non. Pour réaliser cet objectif, la compagnie souhaite utiliser un modèle basé sur un réseau de neurones de 5 couches ayant l'architecture suivante :
- $L^{[0]}$: À trouver par vous-même étant donné la description du problème.
 - $L^{[1]}$: 8 neurones.
 - $L^{[2]}$: 16 neurones.
 - $L^{[3]}$: 8 neurones.
 - $L^{[4]}$: À trouver par vous-même étant donné la description du problème.
1. Représentez graphiquement l'architecture de ce réseau de neurones (la façon dont les neurones sont organisés entre eux).
 2. Donnez les dimensions des différentes matrices et vecteurs X, Z, A, W, b, \hat{Y} de votre réseau. Considérez 50 données en entrée.
 3. Donnez le nombre exact de paramètres devant être appris dans ce réseau.
 4. Donnez les équations de ce réseau pour obtenir une prédiction à partir d'un ensemble de données en entrée. Utilisez la notation matricielle. Veillez à bien définir toutes les fonctions d'activation que vous avez considérées pour chacune des couches.
 5. Proposez une fonction de coût adaptée à cette situation. Indiquez l'expression mathématique de cette fonction.
 6. Voici différentes fonctions d'activations. Parmi ces dernières, indiquez et justifiez lesquelles ne sont pas pertinentes pour être utilisées comme activation pour les neurones d'une couche cachée. Dans ces fonctions, $\kappa \in [0, 1]$ est un hyperparamètre devant être défini par l'utilisateur.

$$\text{ACT-1}(z, \kappa) = \begin{cases} \kappa & \text{if } z > 0 \\ -\kappa & \text{otherwise} \end{cases}$$

$$\text{ACT-2}(z, \kappa) = \begin{cases} z & \text{if } z > 0 \\ \kappa(e^z - 1) & \text{otherwise} \end{cases}$$

$$\text{ACT-3}(z, \kappa) = \begin{cases} \sqrt{\kappa z} & \text{if } z > 0 \end{cases}$$

$$\text{ACT-4}(z, \kappa) = \begin{cases} \kappa z + \kappa^2 & \text{forall } z \end{cases}$$

Solution:

1. 3 neurones pour la couche d'entrée (car 3 features), et 1 neurone pour la couche de sortie (car une seule prédiction)
2. Les matrices sont les suivantes :

$$X = A^{[0]} = n^{[0]} \times m = 3 \times 50$$

$$Z^{[1]} = A^{[1]} = n^{[1]} \times m = 8 \times 50$$

$$Z^{[2]} = A^{[2]} = n^{[2]} \times m = 16 \times 50$$

$$Z^{[3]} = A^{[3]} = n^{[3]} \times m = 8 \times 50$$

$$\hat{Y} = Z^{[4]} = A^{[4]} = n^{[4]} \times m = 1 \times 50$$

$$W^{[1]} = n^{[1]} \times n^{[0]} = 8 \times 3$$

$$W^{[2]} = n^{[2]} \times n^{[1]} = 16 \times 8$$

$$W^{[3]} = n^{[3]} \times n^{[2]} = 8 \times 16$$

$$W^{[4]} = n^{[4]} \times n^{[3]} = 1 \times 8$$

$$b^{[1]} = n^{[1]} \times 1 = 8 \times 1$$

$$b^{[2]} = n^{[2]} \times 1 = 16 \times 1$$

$$b^{[3]} = n^{[3]} \times 1 = 8 \times 1$$

$$b^{[4]} = n^{[4]} \times 1 = 1 \times 1$$

3. On doit compter le nombre de valeurs dans les matrices W et b : 321 paramètres.
4. Pour la dernière couche, on considère la fonction d'activation identité, car on souhaite obtenir une valeur réelle. Pour les couches cachées, n'importe quelle activation non-linéaire (p.e., ReLU) fait l'affaire.

$$A^{[0]} = X^{[l]}$$

for l **in** 1 **to** 4 :

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

$$\hat{Y} = A^{[4]}$$

5. Seulement la deuxième activation est pertinente. La première est une fonction plate, la troisième n'est pas définie dans les valeurs négatives, et la quatrième est linéaire.