

Rapport Projet IA - 2018



Bordeaux INP
ENSC



Table des matières

Résumé du projet	3
Répartition des tâches	3
Questionnaire	4
Illustrations des interfaces & Étapes de l'évaluation	8

Résumé du projet

Notre projet avait pour but la réalisation d'une application de révision des notions d'Intelligence Artificielle : une partie devait se présenter sous forme d'un questionnaire abordant les différentes notions vues en cours, et une autre devait permettre de tester la bonne réalisation d'un algorithme de Dijkstra en pouvant l'essayer à la main soi-même.

Nous avons donc suivi les instructions évoquées et avons divisé notre interface en deux parties : le premier form accueille l'utilisateur sur l'application en lui proposant deux options, soit le questionnaire soit l'algorithme de Dijkstra. Par la suite, il interagira avec un nouveau form, soit l'un soit l'autre en fonction de son choix sur le form d'accueil initial.

Concernant les questions de la partie questionnaire, nous avons choisi de ne présenter que des questions à réponse unique : sur les 4 réponses proposées, une seule est vraie. En contrepartie, nos questions sont plus poussées et méritent une réelle réflexion.

Concernant la partie Dijkstra, nous avons choisi de présenter de façon limpide à l'utilisateur ses choix au fur et à mesure, sur une interface rendant visible sa progression en termes de noeuds. Lorsqu'un noeud est fermé ou ouvert, celui-ci passe simplement d'une liste à droite entrée par l'utilisateur à une liste à gauche représentative de ses anciennes entrées. Nous avons trouvé que cela lui permettrait plus facilement de suivre son raisonnement, qui dans le cas de la recherche du plus court chemin au sens de l'algorithme de Dijkstra, peut parfois être au premier abord contre intuitive. Un bouton au bas de l'interface permet également à l'utilisateur de visualiser à tout moment les réponses attendues sous forme de treeview.

Répartition des tâches

La répartition des tâches a été réalisée dès le début du projet, en fonction de nos capacités et de nos compétences initiales. Si chaque partie à réaliser se voyait confier à un des deux membres de notre groupe de travail, nous avons dès le départ tenu à suivre les avancées de chacun, afin de bien saisir et comprendre les enjeux de chaque tâche. Cela nous a ainsi permis de garder un oeil critique sur le travail de l'autre et d'avancer sur ce projet plus efficacement.

Tâche	Etudiant(s) en charge
Partie I :	
Rédaction des questions	Eva Guihard
Formulaire du questionnaire	Cyril Simonnet
Partie II :	
Dijkstra	Eva Guihard / Cyril Simonnet
Rédaction du rapport	Eva Guihard / Cyril Simonnet

Tableau 1 : répartition des tâches pour le projet IA 2018

Le travail sur l'algorithme de recherche du plus court chemin Dijkstra s'est avéré être la tâche la plus complexe de ce projet. Nous avons donc décidé de travailler de concert sur cette partie, afin de faciliter sa conception et son développement sous Winforms. Nous nous sommes notamment aidés de fonctions déjà existantes (produites pour le cours et le TP).

Questionnaire

Le choix des questions s'est fait selon les supports de cours fournis. Après une première relecture, le questionnaire a été soumis à plusieurs étudiants de 2ème année de l'ENSC afin d'évaluer sa difficulté.

Nous avons obtenu des résultats assez moyens : soit la difficulté du questionnaire est très importante, hypothèse à confirmer avec la passation du questionnaire sur un plus grand nombre d'étudiants ; soit le niveau de connaissances en intelligence artificielle des étudiants était assez bas auquel cas cette tendance devrait se réduire à l'approche des examens.

Pour ce projet, nous avons déterminé 23 questions suivant un modèle à choix multiple, avec 1 seule réponse correcte. Chaque réponse correcte est ci-dessous soulignée et en gras. Pour la passation du questionnaire, 20 questions choisies aléatoirement sont présentées à l'étudiant en vue de sa notation.

Liste des questions

1. La différence entre IA faible et IA forte est d'ordre :
 - a. Technique : l'IA faible est moins aboutie que l'IA forte
 - b. Technique : l'IA faible est plus aboutie que l'IA forte
 - c. **Conceptuelle : ce sont deux façons de concevoir l'IA**
 - d. Il n'y a aucune différence, ces termes désignent la même chose

2. Dans le cas d'un graphe non-orienté, la matrice d'adjacence est :
 - a. Asymétrique
 - b. **Symétrique**
 - c. Ne peut pas être déterminée
 - d. Similaire à celle d'un graphe orienté

3. Dans la théorie des jeux, un des principes de base est :
 - a. **De supposer que l'adversaire joue de façon optimale**
 - b. De supposer que le nombre de coups possibles est fini
 - c. De descendre progressivement dans l'arborescence des coups possibles
 - d. De supposer que l'adversaire va faire des erreurs

4. Comparé à Min-Max, l'algorithme Alpha-Beta :
 - a. N'a rien à voir, ce sont deux méthodes différentes
 - b. Est identique, mais permet d'ajouter des branches optionnelles
 - c. **Est identique, mais permet d'éviter les branches inutiles**

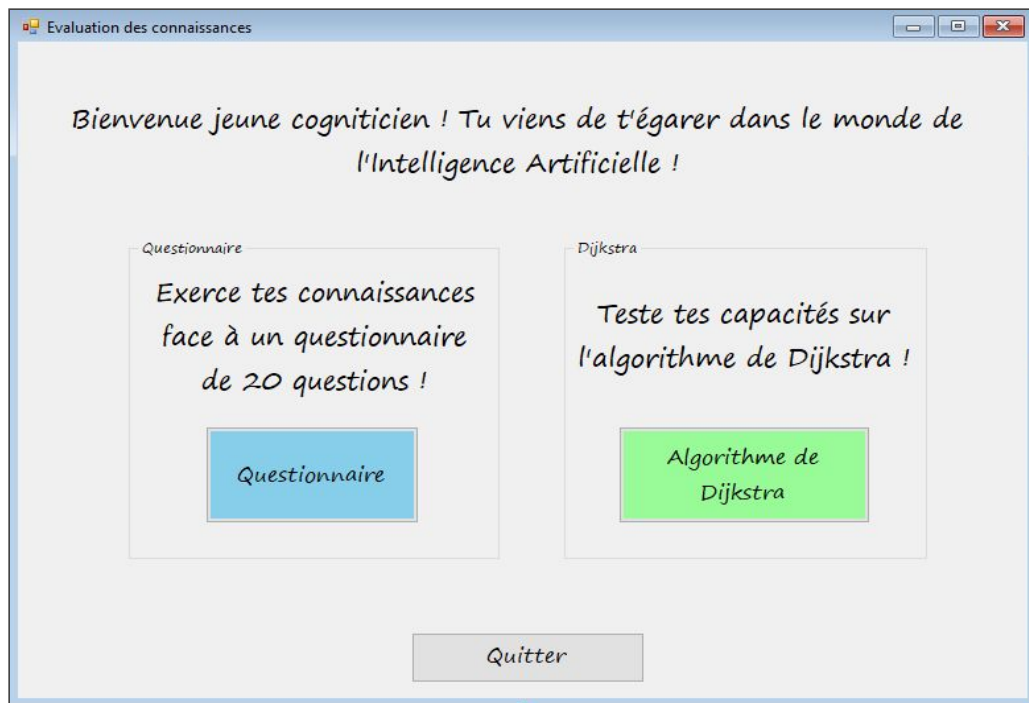
- d. Ne se code pas dans le même langage
5. L'algorithme de Dijkstra permet :
- a. **De trouver le plus court chemin parmi une liste de possibilités finie**
 - b. De déterminer si une liste de possibilité est finie ou infinie
 - c. De trouver le plus court chemin parmi une liste de possibilités infinie
 - d. De valider le fait qu'une liste de possibilité soit finie
6. L'algorithme A* :
- a. Est identique à celui de Dijkstra en tous points
 - b. Est identique à celui de Dijkstra, sauf pour les noeuds fermés
 - c. **Est identique à celui de Dijkstra, sauf pour les noeuds ouverts**
 - d. Diffère entièrement de celui de Dijkstra
7. L'ajout d'une heuristique $h(N)$ dans l'algorithme A* permet :
- a. D'estimer les erreurs que l'IA pourrait commettre
 - b. **D'estimer le coût restant pour atteindre un but fixé**
 - c. De prédire le comportement du système étudié
 - d. Aucune de ces réponses
8. Dans quel cas serait-il pertinent de ne pas choisir de minorant $h(N)$ avec Dijkstra ?
- a. Si trop de paramètres nous sont encore inconnus
 - b. Si l'on souhaite trouver la meilleure solution possible
 - c. Si l'on souhaite exclure la moins bonne solution
 - d. **Si l'on souhaite trouver une solution dans un temps raisonnable**
9. Dans le problème du voyageur de commerce, l'espace des états correspond :
- a. **A l'ensemble des villes et leurs distances entre elles**
 - b. Aux distances entre les villes
 - c. Aux villes elles-mêmes
 - d. On ne peut pas définir d'espace des états pour ce problème
10. Actuellement, l'IA est :
- a. Une simulation de l'intelligence
 - b. De l'intelligence réelle
 - c. Semblable à l'intelligence réelle
 - d. **Aucun consensus n'a encore été validé**
11. Comparé au "model based", le "data based" est :
- a. La même chose, mais en plus abouti
 - b. Basé sur un modèle de connaissance
 - c. **Basé sur un grand nombre de données**
 - d. Complètement différent

12. Quel thème ne fait pas partie du domaine de l'IA aujourd'hui ?
- L'apprentissage symbolique
 - La sémantique**
 - Les arbres de décision
 - La planification
13. Dans une représentation en graphe, un noeud est :
- L'opposé d'un sommet
 - La même chose qu'un sommet**
 - Il n'y a pas de noeuds lorsque l'on parle de représentation en graphe
 - A l'intersection de deux sommets
14. Dans une matrice d'adjacence, lorsqu'il n'y a pas d'arrête :
- On remplace toujours par la valeur 0
 - On remplace toujours par la valeur -1
 - Cela ne peut jamais arriver
 - On remplace par une valeur conventionnelle**
15. Déterminez quelle réponse est erronée.
- Un graphe n'est pas toujours orienté
 - L'algorithme A* est une forme plus aboutie de celui de Dijkstra
 - En IA, tous les problèmes peuvent être rapportés à un algorithme de Dijkstra**
 - Une heuristique permet de réduire les temps de calcul
16. La fonction $f(N)$ se décompose en :
- $\text{coût_chemin}(N) + h(N)$**
 - $\text{coût_chemin}(N) - h(N)$
 - $\text{coût_chemin}(N) / h(N)$
 - $\text{coût_chemin}(N) * h(N)$
17. Si $h(N)$ minore le coût du chemin restant réel, alors :
- L'algorithme ne peut rien conclure
 - A* garantit que le chemin trouvé sera le plus court**
 - La marge d'erreur sera égale à $h(N)$
 - La valeur majorante sera $h'(N)$
18. Enlever une heuristique consiste à :
- Traduire le problème dans un sens plus compréhensible pour la machine
 - Écarter une contrainte du problème pour qu'il devienne plus facile**
 - Déterminer quels points clefs posent le plus problème
 - Retirer du problème une solution évidente pour gagner du temps

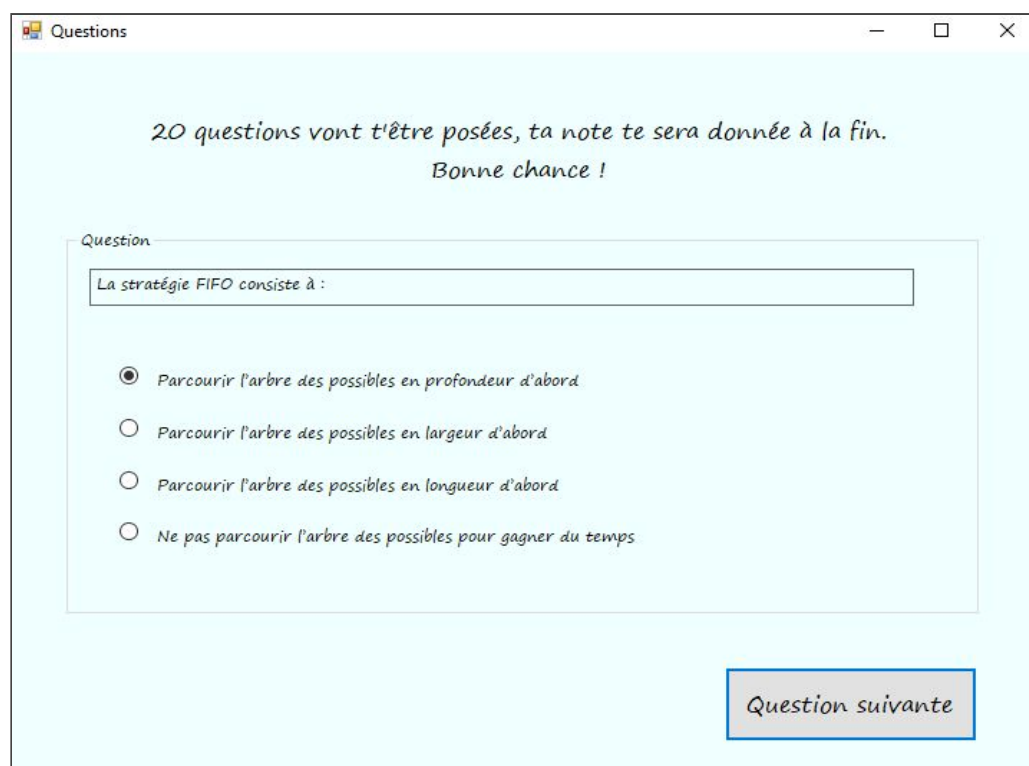
19. L'arbre d'exploration correspond à :
- a. La totalité des possibilités explorées par l'algorithme
 - b. La totalité des possibilités d'exploration de l'algorithme**
 - c. L'espace des états
 - d. L'ordre dans lequel les possibilités vont être explorées
20. Dijkstra est un algorithme à exploration :
- a. Sinusoïdale
 - b. Pyramidale
 - c. Radiale**
 - d. Peut prendre chacune de ces formes
21. Dans le jeu du taquin, l'espace des états peut être représenté par :
- a. Un graphe**
 - b. Une ontologie
 - c. Un réseau bayésien
 - d. Aucune de ces réponses
22. La stratégie FIFO consiste à :
- a. Parcourir l'arbre des possibles en profondeur d'abord
 - b. Parcourir l'arbre des possibles en largeur d'abord**
 - c. Parcourir l'arbre des possibles en longueur d'abord
 - d. Ne pas parcourir l'arbre des possibles pour gagner du temps
23. Dans un problème de maths dont le but serait de trouver un nombre, on aurait en IA :
- a. État initial = l'énoncé, État final = le nombre recherché**
 - b. État initial = le nombre recherché, État final = l'énoncé
 - c. État initial et final = le nombre recherché, en constante évolution
 - d. On ne pourrait pas déterminer ces états

Illustrations des interfaces & Étapes de l'évaluation

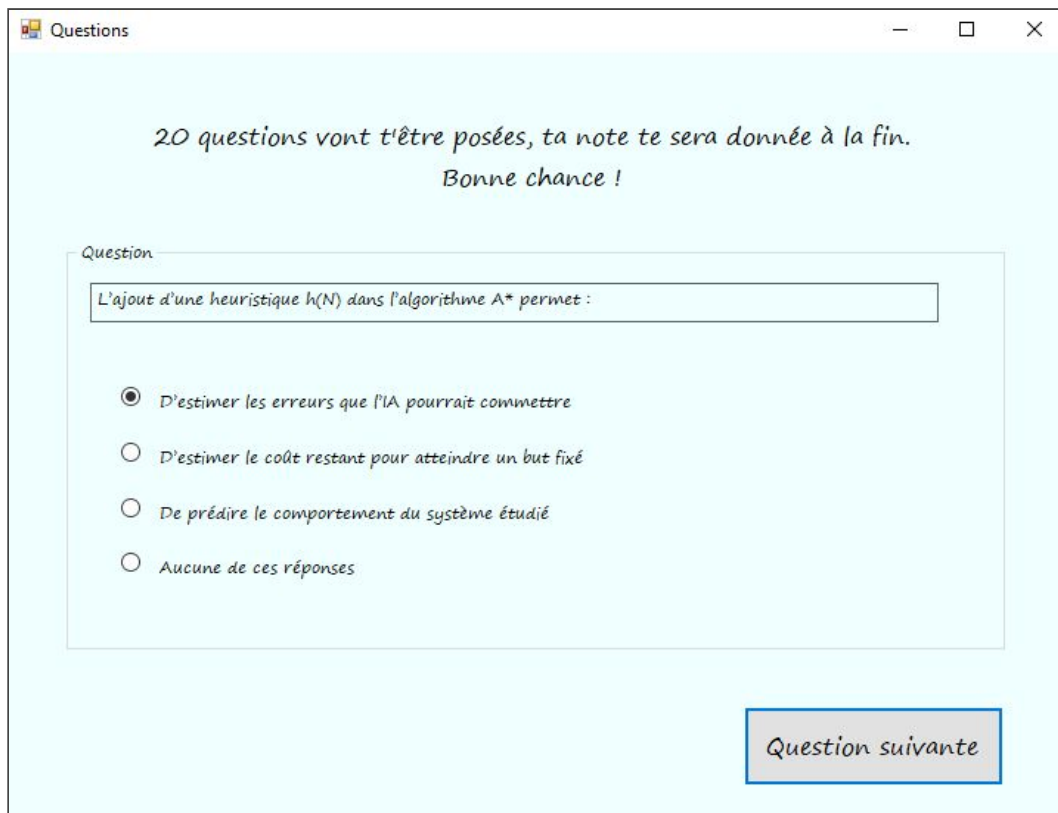
Vous trouverez ci-dessous les étapes illustrées par des captures d'écran de notre interface d'évaluation. Nous commencerons par la partie Questionnaire, puis passerons à la partie Dijkstra. Chaque image sera expliquée par sa légende, en dessous de celle-ci.



1. Premier form : Interface d'accueil de l'application



2. Second form : Interface du Questionnaire



Questions

20 questions vont t'être posées, ta note te sera donnée à la fin.
Bonne chance !

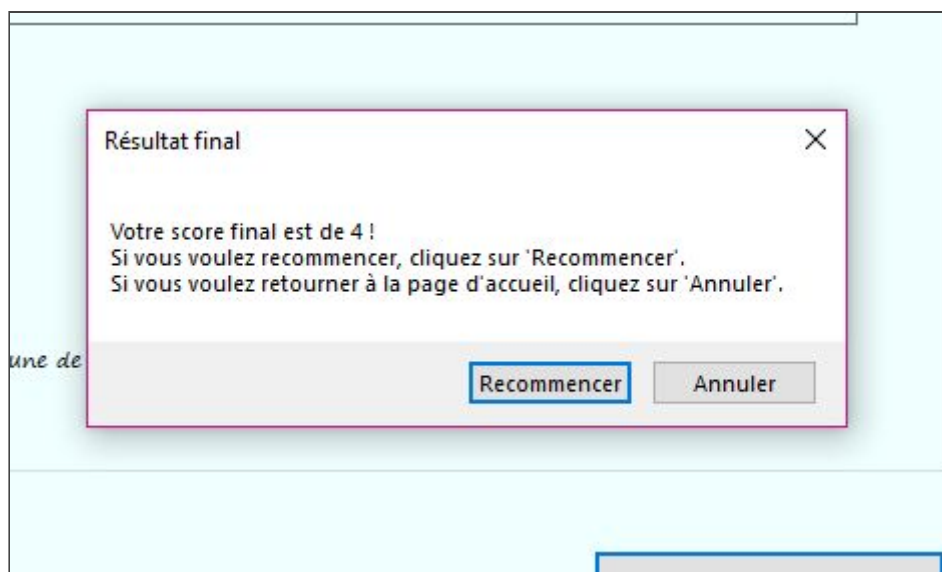
Question

L'ajout d'une heuristique $h(N)$ dans l'algorithme A* permet :

- ☒ D'estimer les erreurs que l'IA pourrait commettre
- ☐ D'estimer le coût restant pour atteindre un but fixé
- ☐ De prédire le comportement du système étudié
- ☐ Aucune de ces réponses

Question suivante

3. Form Questionnaire : exemple d'une autre question



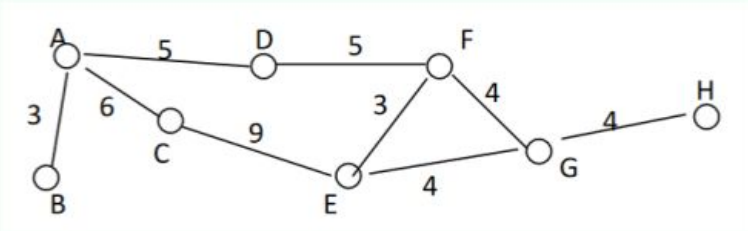
Résultat final

Votre score final est de 4 !
Si vous voulez recommencer, cliquez sur 'Recommencer'.
Si vous voulez retourner à la page d'accueil, cliquez sur 'Annuler'.

Recommencer Annuler

4. Form Questionnaire : à la fin des 20 questions, annonce du résultat et possibilité de recommencer ou de retourner à la page d'accueil

Bienvenue dans cet algorithme de Dijkstra ! Essaye de trouver le meilleur chemin, étape par étape...



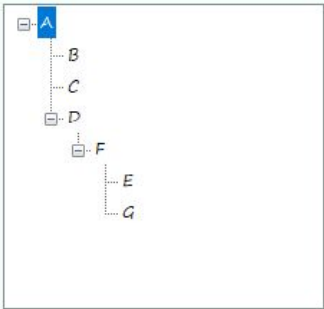
Etape 2 de l'algorithme :

Fermés (précédent)	Fermés
<input type="text"/>	<input type="text"/>
Ouverts (précédent)	Ouverts
<input type="text" value="A"/>	<input type="text"/>

5. Form Dijkstra : après un retour à l'accueil, on ouvre un nouveau form pour le test de Dijkstra

Search Tree Answers

TreeView

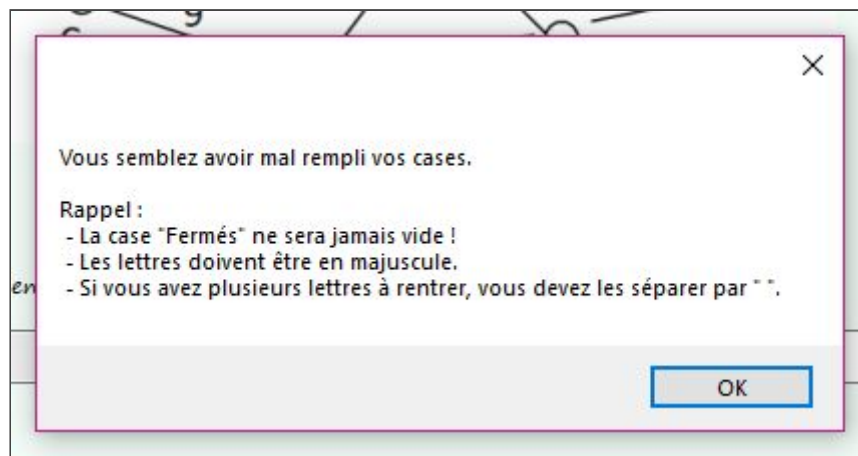


Chemin Final

```

A ---> D : 5
D ---> F : 5
F ---> E : 3
          
```

6. Form Dijkstra : à tout moment, l'utilisateur peut cliquer sur Réponses Vue Search Tree et obtenir les réponses attendues, sous une vue différente



7. Si la saisie d'une étape est incorrecte, ce messageBox s'affichera

Dijkstra

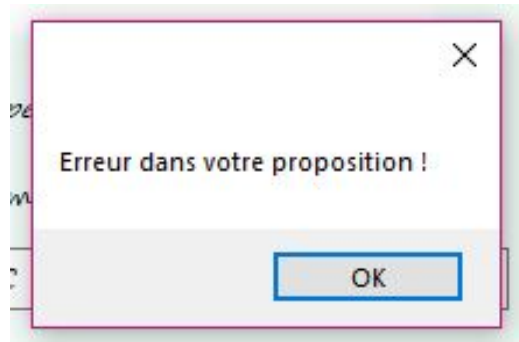
Bienvenue dans cet algorithme de Dijkstra ! Essayez de trouver le meilleur chemin, étape par étape...

Etape 3 de l'algorithme :

Fermés (précédent)	Fermés
<input type="text" value="A"/>	<input type="text" value="A"/>
Ouverts (précédent)	Ouverts
<input type="text" value="B C D"/>	<input type="text" value="B C D"/>

Étape 2 correcte !

8. Form Dijkstra : Etape 3



9. Form Dijkstra : si erreur dans la saisie des lettres, ce messageBox s'affichera

Dijkstra

Bienvenue dans cet algorithme de Dijkstra ! Essayes de trouver le meilleur chemin, étape par étape...

Etape 4 de l'algorithme :

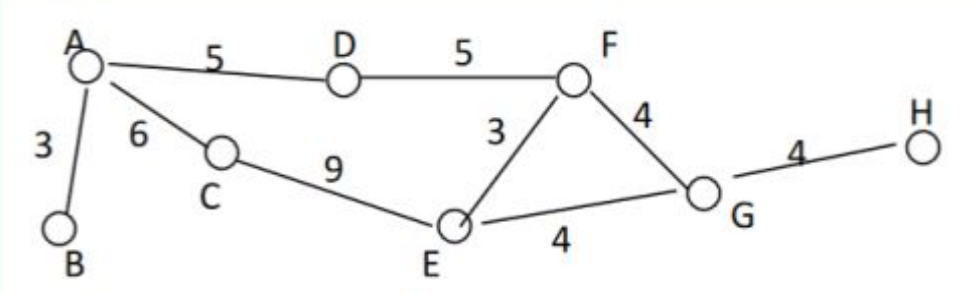
Fermés (précédent)	Fermés
<input type="text" value="A B"/>	<input type="text" value="A B"/>
Ouverts (précédent)	Ouverts
<input type="text" value="C D"/>	<input type="text" value="C D"/>

Étape 3 correcte !

10. Form Dijkstra : Etape 4

Dijkstra
— □ ×

Bienvenue dans cet algorithme de Dijkstra ! Essayes de trouver le meilleur chemin, étape par étape...



Etape 5 de l'algorithme :

Fermés (précédent)

A B D

Ouverts (précédent)

C F

Étape 4 correcte !

Réponses Vue Search Tree

Valider

Fermés

A B D

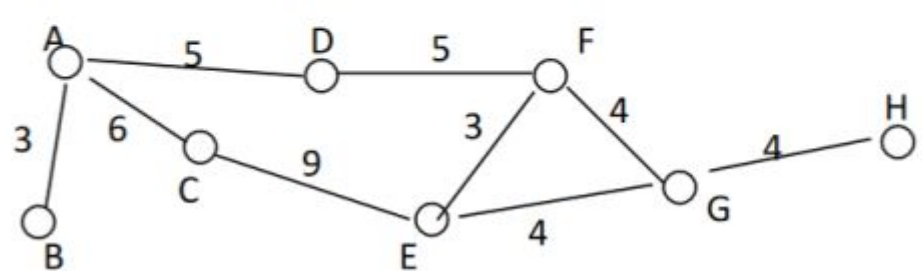
Ouverts

C F

11. Form Dijkstra : Etape 5

Dijkstra
— □ ×

Bienvenue dans cet algorithme de Dijkstra ! Essayes de trouver le meilleur chemin, étape par étape...



Etape 6 de l'algorithme :

Fermés (précédent)

A B D C

Ouverts (précédent)

F E

Étape 5 correcte !

Réponses Vue Search Tree

Valider

Fermés

A B D C

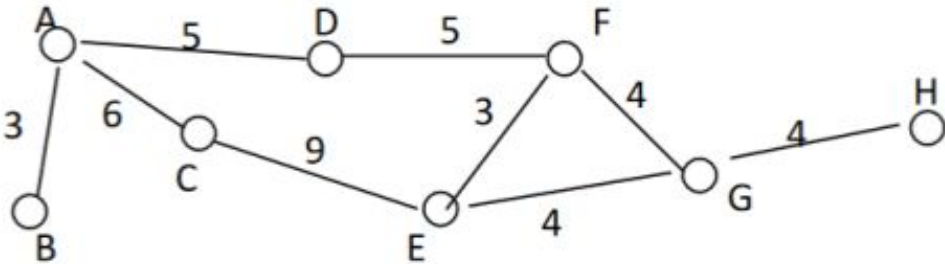
Ouverts

F E

12. Form Dijkstra : Etape 6

Dijkstra
— □ ×

Bienvenue dans cet algorithme de Dijkstra ! Essayes de trouver le meilleur chemin, étape par étape...



Etape 7 de l'algorithme :

Fermés (précédent)

A B D C F

Ouverts (précédent)

E G

Fermés

A B D C F

Ouverts

E G

Étape 6 correcte !

Réponses Vue Search Tree

Valider

13. Form Dijkstra : Etape 7

Bienvenue dans cet algorithme de Dijkstra ! Essayes de trouver le meilleur chemin, étape par étape...

Fermés (précédent)
A B D C F E

Ouverts (précédent)
G

Fin de l'algorithme.
Fermés
A B D C F E

Ouverts
G

Étape 7 correcte !

Réponses Vue Search Tree

Valider

Dijkstra terminé !
OK

14. Form Dijkstra : Fin de l'algorithme