**Exercice 2**

**Méthodes de résolution exactes**

Les méthodes de résolution exactes, aussi appelées méthodes de recherche exhaustives, explorent l’espace des états pour construire les solutions en essayant toutes les combinaisons possibles de manière systémique. Parmi les algorithmes de résolution exacte les plus populaires, nous trouvons :

* **DFS :** Deep First Search ou recherche en profondeur d’abord.
* **BFS :** Breadth First Search ou recherche en largeur d’abord.

Ces algorithmes sont des algorithmes de parcours d’arbres. Durant la construction de l’arbre pour chaque nœud plusieurs branches / chemins sont accessibles. Chaque chemin allant de la racine à une feuille représente une solution potentielle au problème traité. La différence entre DFS et BFS consiste en l’ordre de traitement des nœuds. L’algorithme DFS gère la liste de nœuds candidats comme une *Pile (LIFO)*. A l’inverse de l’algorithme BFS, qui traite la liste de nœuds candidats comme une *File (FIFO)*.

| **Procédure Recherche(G : Graphe )** |
| --- |
| OUVERT : Pile ou File ; /\* Dépend de l’algorithme utilisé DFS ou BFS \*/  FERME : Liste ; /\* Liste des noeuds déjà visités \*/  BestSol : Noeud ; /\* Meilleure courante solution \*/  **Début**  d ←root(G) ;  Insérer d dans OUVERT ;  **Tant que** non Vide(OUVERT) **Faire**  n = Ouvert.noeud( ); /\* Extraire un élément selon la politique FIFO ou LIFO \*/  Insérer n dans FERME ; /\* Marquer comme déjà visité \*/  **Si** EtatFinal(n) **ET** Évaluation(G, n) > Évaluation(G, BestSol)  **Alors** MettreAJour(BestSol) ;  **Fsi;**  nEnfants ←Successeurs(n) ;  **Pour chaque** enfant e de nEnfants **Faire**  Insérer e dans OUVERT ;  **Fait ;**  **Fait ;**  **Fin;** |

**A faire :**

Il vous est demandé d’implémenter l’algorithme de recherche exacte “Depth First Search : DFS” pour la résolution du problème du Voyageur de Commerce.

Testez votre algorithme avec différentes tailles de problèmes. Que remarquez-vous ?

Bon courage.