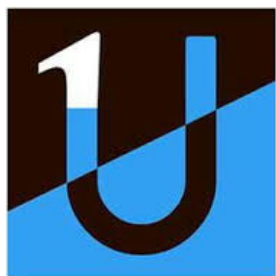


ANNÉE UNIVERSITAIRE 2019-2020
SESSION 1 DE PRINTEMPS



Parcours/Étape: L3 Informatique **Code UE:** 4TIN602U

Épreuve: Techniques Algorithmiques et Programmation

Date: 13/05/2020 **Heure:** 14h00 **Durée:** 1h30

Documents: voir les consignes ci-dessous

Épreuve de M. Cyril GAVOILLE

université
de BORDEAUX

Collège Sciences
et Technologie

C O N S I G N E S

Toutes vos réponses sont à saisir dans le fichier `reponses.txt` accompagnant le sujet, et qui est à déposer sur la page [Moodle](#) du cours où figure également quelques consignes générales. Merci de ne pas renommer ce fichier réponses, ni de le compresser ou de changer son format. Vous pouvez mettre des commentaires directement dans `reponses.txt` si cela vous semble nécessaire. Enfin, ne cherchez pas à me rendre le sujet (.pdf) que je viens de vous envoyer.

Le sujet comporte **3** pages. Vous avez le droit d'utiliser les documents accessibles sur la [page de l'UE](#), en particulier les [notes de cours](#), ainsi que vos propres fichiers (notes et programmes de TP). Il s'agit d'un devoir individuel. Tout constat de manquement à cette consigne sera sanctionné.

Distance de Manhattan

Dans cette partie, on considère des points de la grille à coordonnées entières ainsi que la distance de Manhattan, comme dans le Devoir Maison. Plus précisément, si $A = (x_A, y_A)$ et $B = (x_B, y_B)$ sont deux points de la grille, alors $x_A, y_A, x_B, y_B \in \mathbb{Z}$ et $d(A, B) = |x_A - x_B| + |y_A - y_B|$. On dira que k points u_1, u_2, \dots, u_k sont « alignés » s'il existe un plus court chemin de u_1 à u_k passant par u_1, u_2, \dots, u_k dans cet ordre.

Q1a. *Trouver deux points A, B de la grille tels que l'ensemble formé des 4 points $\{(3, 7), (1, 6), A, B\}$ ne possède pas 3 points alignés. Dit autrement, quels que soient les trois points pris dans l'ensemble, ils ne sont jamais alignés.*

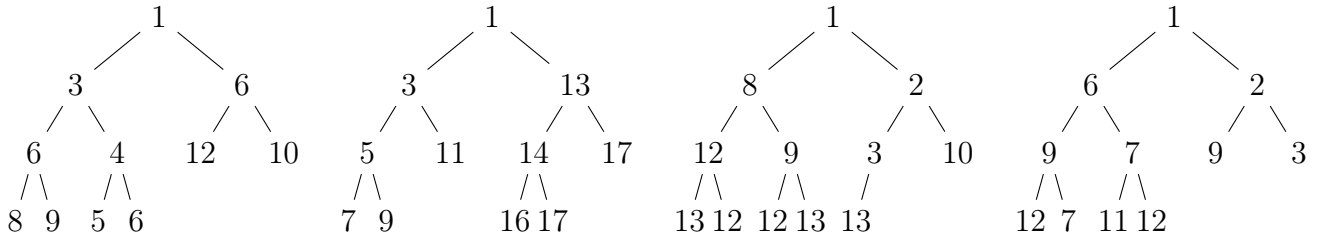
Il est bien connu que tout arbre couvrant de poids minimum pour tout ensemble de points du plan a un degré maximum au plus 6, c'est-à-dire que chaque sommet de l'arbre possède au plus 6 voisins. Il est possible d'obtenir un arbre couvrant de poids minimum avec un sommet de degré 6 en plaçant les points sur les sommets d'un hexagone régulier et un à son centre. Dans cette configuration il est possible d'avoir un arbre couvrant de poids minimum pour ces sept points où le centre est de degré 6, les six autres points étant de degré 1.

Cependant, tout ceci est vrai si le poids des arêtes de l'arbre représente la distance euclidienne entre les points. Pour la distance de Manhattan, c'est différent.

Q1b. *Trouver un ensemble de 9 points de la grille de sorte qu'il possède un arbre couvrant de poids minimum ayant un sommet de degré 8 et où le poids des arêtes de l'arbre correspond à la distance de Manhattan entre les points. Votre ensemble devra contenir les trois points suivants : $(5, 11), (9, 11), (5, 15)$.*

Q1c. *Donnez le poids de l'arbre précédemment construit.*

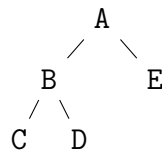
Tas minimum



Q2a. Pour chacun des arbres binaires ci-dessus indiquez ceux correspondant à un tas minimum. (De gauche à droite mettez V si c'est vrai et F sinon.)

Q2b. Donnez le tas résultant de l'insertion dans un tas minimum supposé vide au départ des 10 lettres du mot **GMPRKRFPS** ajoutées une par une en le lisant de gauche à droite. On trie les lettres selon l'ordre alphabétique.

Indiquez la suite des 10 lettres obtenues en lisant l'arbre par niveau, de haut en bas et de gauche à droite. Par exemple, pour l'arbre ci-dessous, il faut indiquer les 5 lettres **A B E C D**.

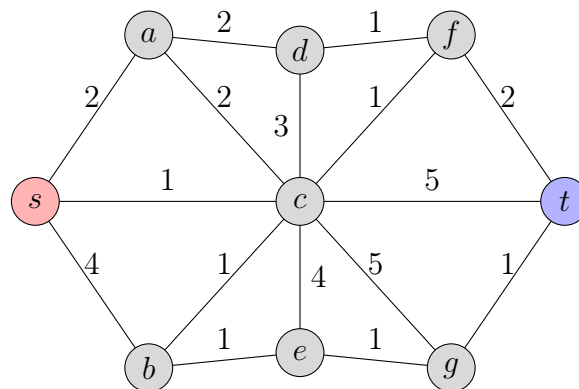


Q2c. Appliquez deux fois le dépilement d'un élément au tas obtenu à la question précédente. Donnez les tas obtenus après l'exécution de chacune des deux opérations (suivant le même format que la question précédente).

Q2d. Quelles sont, dans l'ordre, les deux lettres ainsi supprimées ?

Algorithme A*

On considère le graphe valué suivant.



On souhaite trouver un chemin entre s et t à l'aide de l'algorithme A* vu en cours. Pour cela on va considérer les deux heuristiques h_1, h_2 suivantes, estimant la distance entre un sommet quelconque du graphe et la cible t .

	(s, t)	(a, t)	(b, t)	(c, t)	(d, t)	(e, t)	(f, t)	(g, t)	(t, t)
h_1	0	4	0	4	3	1	2	0	0
h_2	0	5	0	5	3	2	5	11	0

Q3a. Appliquez l'algorithme A^* en utilisant l'heuristique h_1 . Indiquez dans quel ordre les sommets sont ajoutés à P en donnant à chaque fois $\text{parent}[u]$, $\text{coût}[u]$ et $\text{score}[u]$ au moment de l'ajout de u dans P . Notez que t n'est pas ajouté à P par A^* , les informations le concernant ne sont donc pas demandées.

Si durant l'exécution d' A^* il y a un choix, considérez en priorité le plus petit sommet ($a < b < \dots < f < g < s < t$).

Q3b. Donnez le chemin de s à t ainsi construit.

Q3c. Donnez son coût.

Q3d. Mêmes questions en utilisant l'heuristique h_2 .

Q3e. Quelle est la distance entre s et t ?

Q3f. Donnez trois sommets du graphe dont les arêtes valuées forment un triangle ne vérifiant pas l'inégalité triangulaire.

Q3g. Donnez trois sommets du graphe montrant que l'heuristique h_2 n'est pas monotone.