

GRAPHES ET APPLICATIONS

Projet

La guerre des étoiles

Jacques SPONTON/Ahmad AUDI / Mouna EL ATTAR

(audiahmad@hotmail.com)

Ce projet C a pour but de concrétiser vos connaissances relatives au module graphes et applications. Amusez-vous bien ☺

Consignes

- Le projet doit être déposé sur **Moodle** dans le dossier correspondant au numéro de votre classe.
- Le projet à déposer doit être sous la forme d'une archive compressée contenant l'ensemble des fichiers nécessaires à la compilation et à l'utilisation du projet.
- Les critères d'évaluation ont été déposés sur **Moodle**.
- Le code source doit être fourni en C et utilisera les bibliothèques usuelles. Votre code doit pouvoir être compilé, sans erreur et sans warning.
- Le projet peut être développé en binôme ou en trinôme. Toutefois la soutenance sera à titre individuel.
- Le projet doit être accompagné d'un Makefile.
- Le projet doit être modulaire : Création des différents fichiers *.c , *.h et main.c.

Contraintes

- Vous ne devez pas utiliser plus de quatre paramètres dans vos programmes (l'utilisation des structures est fortement recommandée) .
- Il convient, avant chaque programme, d'écrire des commentaires du style qui suit.


```

/*
    But de la fonction
    Explication des paramètres du programme
    Ce que la fonction renvoie
*/
      
```
- Tous les fichiers *.txt doivent être dans un dossier « Data ».

Données

Les données à utiliser seront issues de plusieurs fichiers textes, leur forme est décrite ci-dessous selon chaque type de fichier.

La forme des données peut varier d'un fichier à un autre. En fait, il y a quatre types de fichiers.

Premier type : planeteX.txt / terrainX.txt / bonus.txt

Début : sommetDeDépart (sommet en option selon le fichier à considérer)
 Fin: sommetArrivé (sommet en option selon le fichier à considérer)
 Planetei -> Planetej (nombre réel optionnel selon le fichier à considérer)
 Planetei <- Planete2 (nombre réel optionnel selon le fichier à considérer)
 ...

Deuxième type : typeX.txt

nombreDeTerrain
 typeDeTerraini coutDuTerraini
 ...

Troisième type : typeSommetX.txt

numeroDuSommet typeDeTerrain
 ...

Quatrième type : base.txt

numeroDuSommet poidsDuSommet ValeurNutritiveDuSommet
 ...

Thème

Savez-vous que les graphes sont utilisés dans la planification stratégique et l'analyse de données pour optimiser les résultats de batailles intergalactiques ?

Aussi, bienvenue dans le projet de missions dans l'univers de Star Wars !

Vous serez appelé à diriger une équipe de soldats courageux à travers plusieurs missions à travers la galaxie tout en utilisant les graphes pour élaborer des stratégies efficaces (déplacement d'unités).

Vous aurez 4 missions obligatoires à faire ainsi qu'un bonus pour les plus courageux.

Les parties ne sont pas liées. Toutefois, un même programme peut être utilisé dans plusieurs parties.

Chaque mission contient ses données récupérables sur Moodle.

Que la Force soit avec vous et que vous puissiez remplir chaque mission avec succès !

Développement

Mission 1 :

Vous devez diriger un vaisseau d'une planète à une autre planète. Votre objectif est d'aller de la planète Début à la planète Fin en utilisant successivement les fichiers planete1.txt, planete2.txt et planete3.txt.

- 1) Écrivez un programme qui retourne le chemin permettant d'aller du sommet Début vers le sommet Fin tout en passant par **le moins de sommets**. Le résultat à retourner devra être sous la forme d'un tableau de caractères.

Exemple

Donné : planetej.txt :

Debut: A

Fin: E

A -> B

F <- A

C <- B

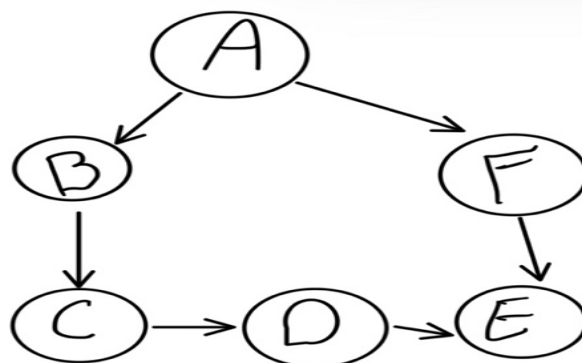
C -> D

D -> E

E <- F

Objectif : Aller de A vers E

Représentation :



Sortie : ['A','F','E'] (On stocke le résultat dans un tableau de caractère et on l'affiche dans le terminal)

2) De nos jours, entretenir les voies de transport dans la galaxie coûte cher, ce qui signifie qu'il y a des autoroutes payantes interplanétaires. Les économies de l'empire n'étant pas infinies, écrivez un programme qui retourne le chemin **le moins coûteux** permettant d'aller du sommet Debut au sommet Fin. Le résultat à retourner devra être sous la forme d'un tableau de caractères. Les fichiers à utiliser successivement sont `planete4.txt`, `planete5.txt` et `planete6.txt`.

Exemple

Donné : `planetei.txt` :

Debut: A

Fin: E

A -> B 10

F <- A 90

C <- B 10

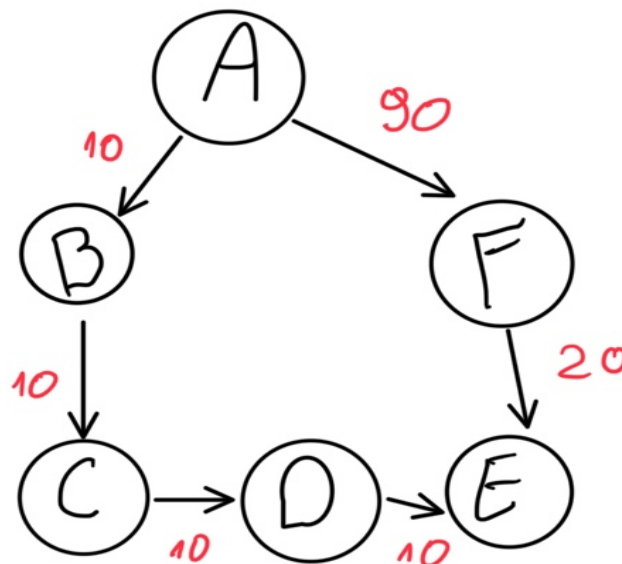
C -> D 10

D -> E 10

E <- F 20

Objectif : Aller de A vers E

Représentation :



Sortie : ['A','B','C','D','E'] (On stocke le résultat dans un tableau de caractère et on l'affiche dans le terminal)

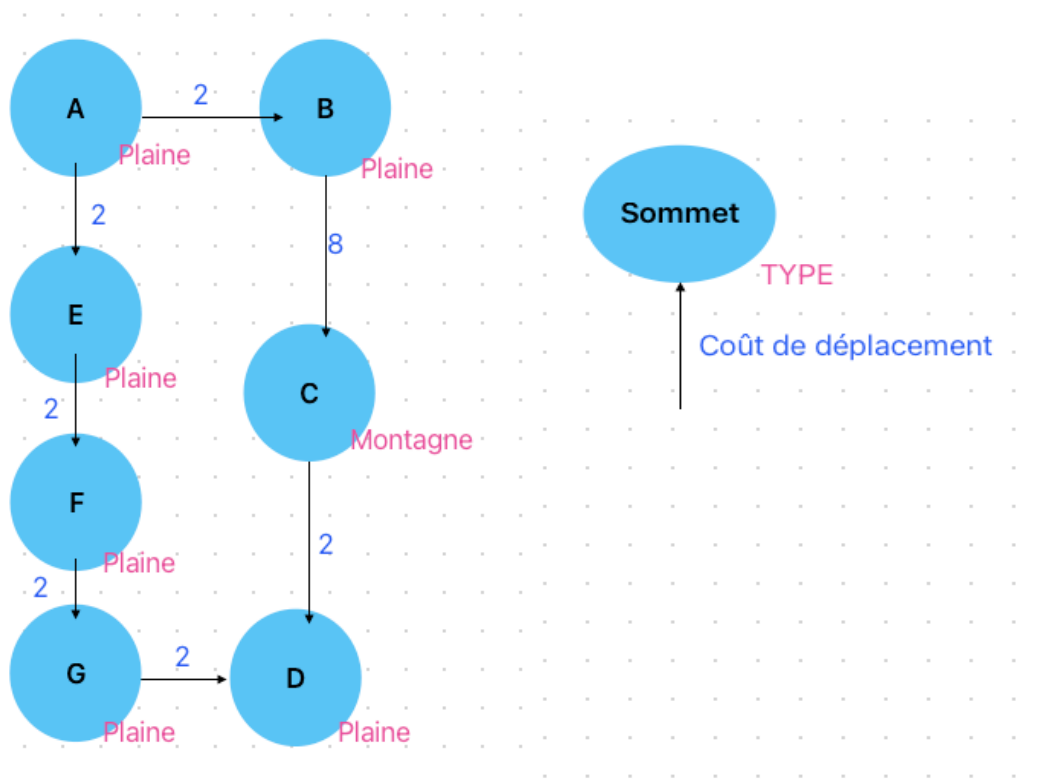
Mission 2 :

Toutes mes félicitations, vous êtes arrivés sur la planète Fin, mais l'aventure ne fait que commencer.

Comme vous le savez, une nouvelle planète dans l'univers de la guerre des étoiles, est égale à un nouveau terrain ainsi que de nouvelles contraintes.

Maintenant, chaque sommet a un type et, à chaque type, correspond un nombre qui représente le coût de déplacement pour accéder à ce type de sommet.

Exemple



Pour aller de $A \rightarrow D$,

2 chemins sont possibles :

Chemin 1 :

$A \rightarrow B \rightarrow C \rightarrow D$

Coût de déplacement : $2 + 8 + 2 = 12$

Chemin 2 :

$A \rightarrow E \rightarrow F \rightarrow G \rightarrow D$

Coût de déplacement : $2 + 2 + 2 + 2 = 8$

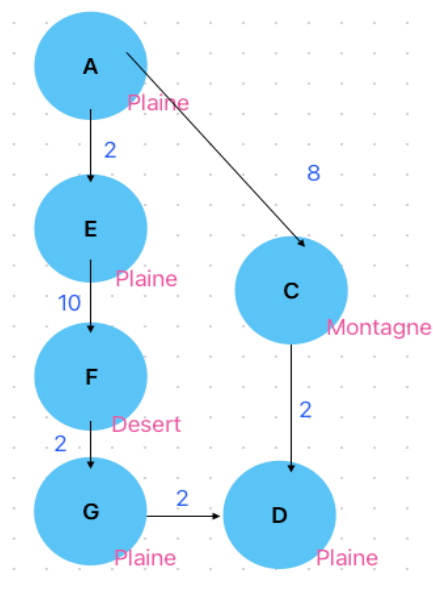
Comme $12 > 8$, on choisit le chemin 2

Vous disposez de 3 unités (homme, voiture et moto) qui ont toutes des missions différentes et avec des contraintes différentes.

Vous commencez par le sommet terrain Debut.

- 3) La moto doit aller du sommet Debut au sommet Fin avec un coût minimum de déplacement et en n'ayant pas le droit d'aller sur les cases Montagne et Marais. En utilisant conjointement les fichiers type1.txt, typeSommet1.txt et terrain1.txt et ensuite les fichiers type2.txt, typeSommet2.txt et terrain2.txt, écrivez le programme qui permet de répondre à ce besoin.

Exemple



Solution : $A \rightarrow E \rightarrow F \rightarrow G \rightarrow D$

Même si ce chemin est plus coûteux, nous devons prendre ce dernier, car la moto ne peut traverser le sommet C (Montagne).

Mission 3 :

Dans votre vaisseau, il y a des voitures et on dispose de deux garages , le garage de départ (sommet source) et le garage de dépôt (sommet puits). On souhaite que les véhicules transitent du garage de départ vers les cinq villes A, B, C, D et E pour ensuite rejoindre sept villes d'arrivée, à savoir, les villes F, G, H, I, J, K, L.

Les capacités de circulation du garage de départ vers les cinq villes sont 70, 30, 40, 55 et 15.

Les capacités de circulation de chacune des sept villes vers le garage de dépôt sont 5, 21, 16, 45, 50, 20 et 7.

Les limitations en matière de circulation sont données par le tableau suivant.

	F	G	H	I	J	K	L
A		30			60		
B	15			35		5	
C					25		50
D		11	12	17	10	4	
E	5	10	2		6		7

- 4) Écrivez le programme permettant de déterminer le nombre maximal de voitures que l'on peut mettre en circulation en même temps pendant une heure.
- 5) Écrivez le programme permettant de déterminer le nombre maximal de voitures que l'on peut mettre en circulation entre tout couple de villes pendant une heure.

Mission 4 :

Le Stormtrooper (un humain) doit ravitailler dans l'urgence les soldats.

Pour ce faire, il doit d'abord récupérer de chaque base une boîte alimentaires scellée et caractérisée par son poids et sa valeur nutritive et ce, tout en minimisant la distance totale parcourue et donc, le temps de récupération des différentes boîtes.

- 6) Écrivez le programme qui permet de répondre à ce besoin. A cette fin vous utiliserez le fichier terrain3.txt.

Maintenant, sachant que le Stormtrooper a récupéré toutes les boîtes en un temps minimal, il doit décider des boîtes à transporter de manière à maximiser la valeur nutritive totale des boîtes choisies et tout en respectant la contrainte de charge maximale qu'il peut transporter et qui est de 40 kg.

- 7) Écrivez le programme qui permet de répondre à ce besoin. A cette fin vous utiliserez le fichier base.txt.

Extension (Bonus)

Pour des raisons de planification, le Stormtrooper a eu pour ordre de récupérer la première boîte à partir de la base Début, de passer par toutes les autres bases une et une seule fois afin de récupérer les autres boîtes et de terminer son parcours en revenant à la base de départ, à savoir, la base Début et ce, tout en minimisant la distance totale parcourue.

- 8) Écrivez le programme qui permet de répondre à ce besoin. A cette fin, on donnera le programme qui se base sur une méthode de résolution approchée tout en utilisant le fichier bonus.txt.