

ORDONNANCEMENT

*Prenez le temps de lire attentivement ce document. Il contient notamment certaines instructions que vous devez absolument respecter. Tout manquement influencera fatalement votre note.
D'autres éléments d'informations (clarification, complément) pourront vous être fournis ultérieurement.*

PROGRAMME A DEVELOPPER

Votre programme se déroule en plusieurs étapes :

1. Lecture d'un tableau de contraintes contenu dans un fichier .txt, mise en mémoire et affichage de ce tableau sur l'écran ;
2. Construction d'un graphe correspondant à ce tableau de contraintes ;
3. Vérification du fait que ce graphe possède toutes les propriétés nécessaires pour qu'il soit un graphe d'ordonnement :
 - Un seul point d'entrée
 - Un seul point de sortie
 - Pas de circuit
 - Valeurs identiques pour tous les arcs incidents vers l'extérieur à un sommet,
 - Arcs incidents vers l'extérieur au point d'entrée ont une valeur nulle,
 - Pas d'arcs à valeur négative.
4. Si toutes ces propriétés sont vérifiées, calculer le calendrier au plus tôt, le calendrier au plus tard et les marges.
Pour le calcul du calendrier au plus tard, utilisez la convention que la date au plus tard de fin de projet soit égale à sa date au plus tôt. Comme vous savez, pour le calcul des calendriers il faut d'abord effectuer le tri topologique du graphe : ordonner les sommets dans l'ordre des rangs croissants. Il faut donc affecter un rang à chaque sommet, en utilisant un algorithme de votre choix parmi ceux que vous avez vu en cours.

Organisation du travail

Travail par équipes

Le nombre d'étudiants par équipe : normalement, 5 ; il y aura quelques équipes de 4, en fonction du nombre total d'élèves dans votre groupe.

Constitution des équipes : à remettre à votre enseignant le 2 mars au plus tard.

Aucun changement ne sera possible après cette date. A défaut d'une constitution des équipes fournie par les délégués de chaque groupe TD, les enseignants pourront décider eux-mêmes de la constitution des équipes, sans possibilité de modification.

Langage de programmation

Au choix : C, C++, Python, Java

Le langage choisi doit cependant être suffisamment maîtrisé par tous les membres d'une même équipe afin que tous puissent participer. Durant la soutenance, votre enseignant pourra poser n'importe quelle question à n'importe quel membre de l'équipe.

Votre programme doit pouvoir être compilé / exécuté par votre enseignant, sur son PC. Il vous indiquera ce dont il dispose. Vous devrez vous y adapter.

Soutenances

Présentation 15 minutes + démonstration de votre programme + questions/réponses.

Les soutenances ont une durée limitée. Votre enseignant a un planning très serré. Il ne pourra pas continuer les soutenances après le créneau horaire prévu.

Il vous est donc vivement conseillé d'être vraiment prêt à l'heure du début de votre soutenance, ce qui implique (au cas où une soutenance se déroule en présentiel) :

- attendre à la porte de la salle et entrer dès que c'est à votre tour ;
- avoir préparé votre ordinateur, y avoir inclus tous vos programmes et fichiers contenant les tableaux de contraintes ;
- avoir vérifié le chargement de la batterie de l'ordinateur ;
- l'avoir démarré et mis en mode veille ;
- avoir un ordinateur de secours sur lequel vous avez aussi vérifié le bon fonctionnement de votre programme, au cas où...

Tous les ans, il y a plein d'étudiants qui pensent ne jamais avoir de problème. Tous les ans, il y en a qui en ont. Résultat : ils sont stressés et ont moins de temps pour leur soutenance. Dommage...

Tableau de contraintes de test

Des tableaux de test vous seront fournis avant la soutenance sous forme de fichiers .txt, dans un délai suffisant pour vous permettre de les tester avec votre programme. Ces fichiers doivent être disponibles sur votre ordinateur au moment de la soutenance.

N'attendez pas l'arrivée des tableaux de contraintes de test pour tester votre programme ! Il serait alors trop tard.

N'hésitez pas pendant votre développement à utiliser tous les tableaux de contraintes vus en cours et en travaux dirigés, et bien d'autres encore. Plus vous ferez de tests, plus vous pourrez être certains que votre programme fonctionne correctement.

Rendu du travail :

Toutes les équipes devront remettre leur travail à leur enseignant dimanche le 12 avril au plus tard.

Le contenu du rendu :

- Code source : Tout fichier code **que vous avez tapé vous-même** à l'exclusion de tout autre fichier (**donc aucun fichier produit par le logiciel durant la compilation ou exécution**), bien commenté.
- Tous les fichiers .txt des tableaux de test (oui, malgré le fait que ce sont vos enseignants qui vous ont fourni les tableaux de test), **dans le même répertoire que le code.**
- Un ppt ou pdf de la présentation. (Ce fichier peut éventuellement n'être demandé qu'à la soutenance).
- Les traces d'exécution sous forme de fichiers .txt, un fichier par tableau de test. Vous devrez exécuter votre programme sur l'intégralité des tableaux de test et fournir les traces d'exécution correspondantes.
Un tableau de contraintes non testé, ou pour lequel les traces d'exécution ne seront pas fournies, sera considéré comme un cas sur lequel votre programme ne fonctionne pas correctement.

Tout fichier que vous utilisez (et transmettez à votre enseignant) doit être préfixé par votre numéro d'équipe : par exemple, si vous avez un fichier « main » et si vous utilisez le langage C++, et que vous êtes dans l'équipe B2, ce fichier doit avoir le nom B2-main.cpp (ou B2_main.cpp).

Tableaux de contraintes à prendre en compte

Un tableau de contraintes aura une forme suivante (on prend le tableau C01 de l'annexe comme exemple, et on a remplacé les étiquettes des tâches alphabétiques par des nombres ($A \rightarrow 1$, $B \rightarrow 2$ etc.) et où sur chaque ligne le premier chiffre est le numéro de tâche, le deuxième sa durée, et les autres chiffres, si présents, sont des contraintes (prédécesseurs)) :

```

1 9
2 2
3 3 2
4 5 1
5 2 1 4
6 2 5
7 2 4
8 4 4 5
9 5 4
10 1 2 3
11 2 1 5 6 7 8

```

Les N tâches sont numérotées de 1 à N . La tâche fictive α sera notée 0. La tâche fictive ω aura le numéro $N+1$.

Votre programme doit être capable d'importer un tableau de contraintes quelconque répondant aux critères ci-dessus, y compris au cas où le graphe correspondant à ce tableau contiendra des circuits et/ou ne sera pas connexe, et le transformer en un graphe sous forme matriciel (une matrice d'adjacence + une matrice de valeurs).

Fonctions à mettre en œuvre

Déroulement du programme

Mettre en place un programme qui exécute les actions suivantes préalables à l'ordonnancement :

1. Lecture d'un tableau de contraintes donné dans un fichier texte (.txt) et stockage en mémoire
2. Affichage du graphe correspondant sous forme matricielle (matrice d'adjacence + matrice des valeurs). Attention : cet affichage doit se faire à partir du contenu mémoire, et non pas directement en lisant le fichier.
3. Vérifier toutes les propriétés nécessaires du graphe pour qu'il puisse servir d'un graphe d'ordonnancement :
 - un seul point d'entrée,
 - un seul point de sortie,
 - pas de circuit,
 - valeurs identiques pour tous les arcs incidents vers l'extérieur à un sommet,
 - arcs incidents vers l'extérieur au point d'entrée de valeur nulle,
 - pas d'arcs à valeur négative.

Si la réponse à la question 3 est « Oui », procéder au calcul des calendriers :

4. Calculer les rangs de tous les sommets du graphe.
5. Calculer le calendrier au plus tôt, le calendrier au plus tard et les marges.
Pour le calcul du calendrier au plus tard, considérez que la date au plus tard de fin de projet est égale à sa date au plus tôt.

Lors de son exécution, afin de faciliter le déroulement de votre soutenance, votre programme doit être capable de « boucler » sur une série de tableaux de contraintes que vous aurez préparées. Stopper votre programme après chaque graphe puis le relancer n'est pas une bonne solution, et cela entraînera une pénalité.

La structure globale de votre programme est illustrée par le pseudo-code suivant :

La structure globale de votre programme est illustrée par le pseudo-code suivant :

```

Début
    Tant que l'utilisateur décide de tester un tableau de
    contraintes faire
        Choisir le tableau de contraintes à traiter
        Lire le tableau de contraintes sur fichier et le
        stocker en mémoire
        Créer les matrices correspondant au graphe
            représentant ce tableau de contraintes et les
            afficher
        Vérifier que les propriétés nécessaires pour que ce
        graphe soit un graphe d'ordonnancement sont
        vérifiées
        SI oui alors
            calculer les rangs des sommets et les afficher
            calculer les calendriers au plus tôt et au plus
            tard et les afficher
            calculer les marges et les afficher
            calculer le(s) chemin(s) critique(s) et les
            afficher
        finsi
        sinon proposer à l'utilisateur de changer de tableau
        de contraintes
    fait
Fin
  
```

Les 5 étapes doivent être mises en œuvre individuellement.

Il est bien évident qu'il est possible de mettre en œuvre la détection de circuit dans l'algorithme de calcul de rang. Le but du TP/Projet est de vous faire apprendre le plus d'algorithmes « clés » de la théorie des graphes, et donc nous vous l'interdisons.

Traces d'exécution

Chacune des 4 étapes doit afficher des traces du déroulement de l'algorithme. En voici un exemple qui correspond à un tableau de contraintes suivant :

```

1 1
2 2
3 3 1
4 4 1 2
5 5 2 4
  
```

Nous donnons un exemple d'affichage pour les trois premières étapes, les autres étapes doivent suivre le même principe de lisibilité.

Etape	Exemple de trace (ce ne sont que des exemples : vous pouvez faire ce que vous voulez, pourvu que l'on comprenne vite et sans problème comment votre algorithme fonctionne)
1	<p><i>Affichage à chaque ligne du fichier en entrée de ce qui a été lu, par exemple (voir annexe 1 pour la structure du fichier) :</i></p> <p>* Création du graphe d'ordonnement :</p> <pre> 7 sommets 9 arcs 0 -> 1 = 0 0 -> 2 = 0 1 -> 3 = 1 1 -> 4 = 1 2 -> 4 = 2 2 -> 5 = 2 3 -> 6 = 3 4 -> 5 = 4 5 -> 6 = 5 </pre>
2	<p>Représentation du graphe sous forme matricielle</p> <p>Matrice d'adjacence</p> <pre> 0 1 2 3 4 5 6 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 2 0 0 0 0 0 1 1 0 3 0 0 0 0 0 0 0 1 4 0 0 0 0 0 0 1 0 5 0 0 0 0 0 0 0 1 6 0 0 0 0 0 0 0 0 </pre> <p>Matrice des valeurs</p> <pre> 0 1 2 3 4 5 6 0 * * 0 0 * * * * 1 * * * 1 1 * * * 2 * * * * 2 2 * * 3 * * * * * * 3 * 4 * * * * * 4 * * 5 * * * * * * 5 * 6 * * * * * * * * </pre> <p>(Veillez à ce que les matrices soient affichées avec des colonnes bien alignées et avec les entêtes des lignes et colonnes.)</p>
3	<p>Il y a un seul point d'entrée, 0 Il y a un seul point de sortie, 6</p> <p>Détection de circuit (<i>Par exemple, avec la méthode de suppression des points d'entrée</i>) :</p> <p>* Détection de circuit</p> <p>* Méthode d'élimination des points d'entrée</p> <p>Points d'entrée : 0</p> <p>Suppression des points d'entrée</p> <p>Sommets restant : 1 2 3 4 5 6</p> <p>Points d'entrée : 1 2</p> <p>Suppression des points d'entrée</p> <p>Sommets restant : 3 4 5 6</p> <p>Points d'entrée : 3 4</p> <p>Suppression des points d'entrée</p>

	Sommets restant : 5 6 Points d'entrée : 5 Suppression des points d'entrée Sommets restant : 6 Points d'entrée : 6 Suppression des points d'entrée Sommets restant : Aucun -> Il n'y a pas de circuit Les valeurs pour tous les arcs incidents vers l'extérieur à un sommet sont identiques Les arcs 0->1 et 0->2 sont nuls Il n'y a pas d'arcs négatifs -> C'est un graphe d'ordonnement
--	---

Communication

Votre enseignant vous donnera l'adresse email que vous pourrez utiliser pour l'envoi des fichiers ainsi que pour toute question.

Tout email doit avoir comme préfixe du sujet la chaîne TG-PRJ-<numéro d'équipe>, éventuellement suivie de l'objet de votre email. Par exemple, lors du rendu, l'équipe A5 pourra envoyer un email dont le sujet est « TG-PRJ-A5 Rendu final ».

Ces consignes sont très importantes et leur violation même partielle risque d'entraîner des pénalités. Gardez en mémoire que vos enseignants peuvent mettre en place des traitements automatiques lors de la réception de vos emails, et donc que toute défaillance de votre part peut entraîner un mauvais traitement de votre travail (cela peut aller jusqu'à la mise en spam de vos emails...), avec toutes les conséquences que cela peut entraîner.