

## ORDONNANCEMENT

*Prenez le temps de lire attentivement ce document. Il contient notamment certaines instructions que vous devez absolument respecter. Tout manquement influencera 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 que ce graphe ne possède pas de circuit et qu'il n'y a pas d'arcs à valeur négative.
4. Si ces propriétés sont vérifiées, calculer le calendrier au plus tôt, la durée totale de projet, le calendrier au plus tard, les marges et le(s) chemin(s) critique(s).  
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 et/ou en TD.

### Organisation du travail

#### Travail par équipes

Le nombre d'étudiants par équipe : normalement, 4 ou 5, en fonction du nombre total d'élèves dans votre groupe (pas plus que 8 équipes par groupe).

Constitution des équipes : à remettre sur Moodle sous la forme d'un fichier Excel par **le délégué de groupe** (une feuille Excel par groupe) **le 1 mars** au plus tard.

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.

#### 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.**

#### Soutenances

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

Tous les membres d'équipe doivent prendre part dans la soutenance de façon suffisamment égale.

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 **le 21 mars** sous forme de fichiers .txt 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 sur Moodle **dimanche le 30 mars** au plus tard (les dépôts seront fermés 31/03 à 00h.)

#### *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.
- Un pptx 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. Attention, les copies d'écran ne sont pas acceptées.  
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.

#### **Tableaux de contraintes à prendre en compte**

Un tableau de contraintes aura une forme suivante (on prend le tableau C01 de Feuilles d'exercices comme exemple, et on a remplacé les étiquettes des tâches alphabétiques par des nombres (A→1, B→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  $\alpha$  sera notée 0. La tâche fictive  $\omega$  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 le transformer en un graphe sous forme matriciel (une matrice de valeurs).

**Fonctions à mettre en œuvre (pour avoir au moins 10, il faut que les points 1-4 soient opérationnels)**

*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 des valeurs). Attention : cet affichage doit se faire à partir du contenu mémoire, et non pas directement en lisant le fichier. Ce graphe doit incorporer les deux sommets fictifs  $\alpha$  et  $\omega$  (notés 0 et N+1 où N est le nombre de tâches).
3. Vérification des propriétés nécessaires du graphe pour qu'il puisse servir d'un graphe d'ordonnancement :
  - pas de circuit,
  - 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.
6. Calculer le(s) chemin(s) critique(s) et les afficher

*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 :

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 la matrice correspondant au graphe représentant ce tableau de contraintes et l'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

Sinon

Proposer à l'utilisateur de changer de tableau de contraintes

fin Tant que

Fin

Remarque : Il est bien évident qu'il est possible de mettre en œuvre la détection de circuit dans l'algorithme de calcul de rang. Vous n'affichez les rangs que dans l'absence de circuit.

#### Traces d'exécution

Chacune des 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.

Étape	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	<i>Affichage du graphe comme un jeu de triplets , par exemple:</i>

	<p>* Création du graphe d'ordonnancement :</p> <p>7 sommets</p> <p>9 arcs</p> <p>0 -&gt; 1 = 0</p> <p>0 -&gt; 2 = 0</p> <p>1 -&gt; 3 = 1</p> <p>1 -&gt; 4 = 1</p> <p>2 -&gt; 4 = 2</p> <p>2 -&gt; 5 = 2</p> <p>3 -&gt; 6 = 3</p> <p>4 -&gt; 5 = 4</p> <p>5 -&gt; 6 = 5</p>																																																																
2	<p>Matrice des valeurs</p> <table><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>0</td><td>*</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td>*</td></tr><tr><td>1</td><td>*</td><td>*</td><td>*</td><td>1</td><td>1</td><td>*</td><td>*</td></tr><tr><td>2</td><td>*</td><td>*</td><td>*</td><td>*</td><td>2</td><td>2</td><td>*</td></tr><tr><td>3</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>3</td></tr><tr><td>4</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>4</td><td>*</td></tr><tr><td>5</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>5</td></tr><tr><td>6</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr></table> <p>(Veillez à ce que la matrice soit affichée avec des colonnes bien alignées et avec les entêtes des lignes et colonnes.)</p>		0	1	2	3	4	5	6	0	*	0	0	*	*	*	*	1	*	*	*	1	1	*	*	2	*	*	*	*	2	2	*	3	*	*	*	*	*	*	3	4	*	*	*	*	*	4	*	5	*	*	*	*	*	*	5	6	*	*	*	*	*	*	*
	0	1	2	3	4	5	6																																																										
0	*	0	0	*	*	*	*																																																										
1	*	*	*	1	1	*	*																																																										
2	*	*	*	*	2	2	*																																																										
3	*	*	*	*	*	*	3																																																										
4	*	*	*	*	*	4	*																																																										
5	*	*	*	*	*	*	5																																																										
6	*	*	*	*	*	*	*																																																										
3	<p>Il y a un seul point d'entrée, 0</p> <p>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> <p>Sommets restant : 5 6</p> <p>Points d'entrée : 5</p> <p>Suppression des points d'entrée</p> <p>Sommets restant : 6</p> <p>Points d'entrée : 6</p> <p>Suppression des points d'entrée</p> <p>Sommets restant : Aucun</p> <p>-&gt; Il n'y a pas de circuit</p> <p>Il n'y a pas d'arcs négatifs</p> <p>-&gt; C'est un graphe d'ordonnancement</p>																																																																