

ENSEEIHT

GRAPHES

PROJET

---

# Ordonnancement parallèle de tâches sur un graphe orienté acyclique

---

*auteurs :*

BARRAS - THOMAS

MAUREL - SIMON

# Table des matières

0.1	Question 1 . . . . .	1
0.2	Question 2 . . . . .	1
0.3	Question 3 . . . . .	1
0.4	Question 4 . . . . .	1
0.5	Question 5 . . . . .	2

## 0.1 Question 1

- (a) Les noeuds du DAG représentent l'existence d'un sommet du maillage.
- (b) La condition "un noeud  $v_1$  doit être traité avant un noeud  $v_2$ " est : un chemin reliant  $v_2$  et  $v_1$  qui respecte l'orientation.

## 0.2 Question 2

- (a) Pour inclure le coût des communications, on pondère les arêtes.
- (b) L'équilibrage de charges se traduit par le fait que l'arbre a une profondeur faible.  
Le volume de communications réduit se traduit par le fait que la pondération totale des arcs reliant 2 partitions soit minimale.
- (c) bonus

## 0.3 Question 3

- (a) Lorsqu'une faute est commise, il faut recalculer toutes les tâches parentes menant à  $v_i$  car les résultats sont chaînés ( ils dépendent des résultats précédents ) et non conservés.
- (b) En supposant que nous détectons immédiatement l'erreur, et que l'exécution s'arrête à ce moment, puisque tous les résultats précédents sont enregistrés, seule la tâche où s'est produite l'erreur est réexécutée, on a donc  $\mu = 1$
- (c) On commence par calculer le nombre de combinaisons possibles pour créer l'ensemble C, dans le cas où s tâches sont sauvegardées. Dans ce cas là, on a alors  $\binom{n}{s}$  ensembles de sauvegarde différents possibles. De là, pour chaque ensemble, on va tenter de regarder le nombre d'arcs à parcourir pour corriger l'erreur. Dans le pire des cas, on aura alors tous les arcs à parcourir, ce qui fait donc au total :  $m \cdot \binom{n}{s}$
- (d)

## 0.4 Question 4

- Lorsqu'il n'y a pas de parallélisme, il faut parcourir toutes les tâches séquentiellement, en respectant l'ordre des dépendances. Le temps d'exécution est donc égal au nombre de tâches du DAG.

## 0.5 Question 5

(a) états :

- X = non numéroté et avec certains de ses prédécesseurs non numérotés
- Y = non numéroté et avec tous ses prédécesseurs numérotés
- Z = numéroté

L'algorithme parcourt le graphe pour numéroté les noeuds. La propriété qui garantit l'ordre partiel du graphe est que pour numéroté un noeud, il faut que tous ses prédécesseurs le soient. C'est l'ensemble intermédiaire Y qui permet de s'assurer que lorsque l'on numérote une tâche, tous ses prédécesseurs sont forcément numérotés. La propriété qui garantit que l'ordre est total est l'unicité du numéro attribué aux tâches.

(b) (TODO : On parcourt n fois la boucle, et pour chaque itération dans la boucle on effectue  $c(\text{succ}) + c^*(m/n)$  (prec pour chaque succ, en considérant que le nombre moyen de predecesseur pour un noeud est  $m/n$ ) soit  $c^*(n+m)$  )

## 0.6 Question 6

- Les parcours induits par l'utilisation d'une pile sont les parcours en profondeur.  
Les parcours induits par l'utilisation d'une file sont les parcours en largeur.

## 0.7 Question 8

- (a) Dans le meilleur des cas on utilise pleinement les ressources donc la borne inférieure du temps total d'exécution est :  $\frac{n}{r}$   
Pour atteindre cette borne il faut utiliser toutes les ressources à chaque étape sauf pour la dernière.
- (b) La contrainte des ressources limitées se traduit par le fait qu'à chaque étape, il y a plus de ressources que de noeuds.

## 0.8 Question 10

## 0.9 Question 11

## 0.10 Question 13

## 0.11 Question 14

## 0.12 Question 16