

4. Modélisation d'un problème de génération de code

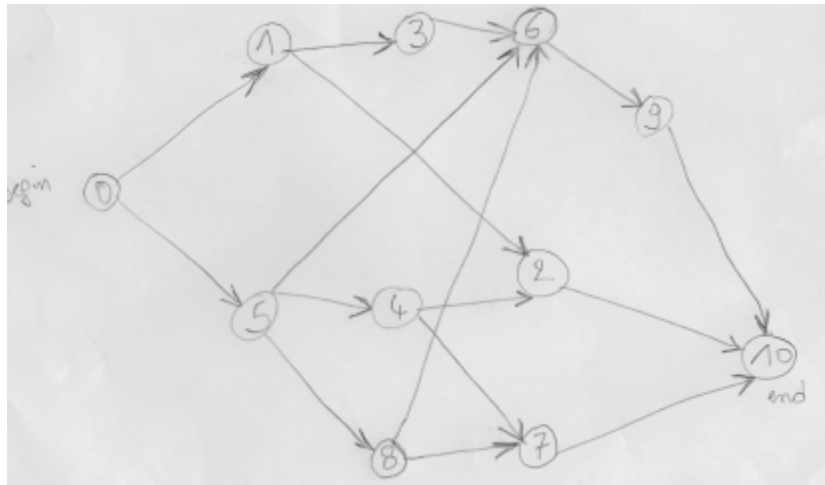
(3 pts)

On considère l'ensemble suivant d'instructions (numérotées de 1 à 9) que l'on doit exécuter sur un seul et unique processeur.

- | | |
|-----------------|---------------------|
| 1. read(a) | 6. $f := h + c / e$ |
| 2. $b := a + d$ | 7. $g := d * h$ |
| 3. $c := 2 * a$ | 8. $h := e - 5$ |
| 4. $d := e + 1$ | 9. $i := h - f$ |
| 5. read(e) | |

Q1. Représentation sous forme de graphe.

- Quelle instruction doit immédiatement précéder l'instruction #8 ? *→ la #5*
- Quelles instructions doivent immédiatement précéder l'instruction #7 ? *→ la #4 et la #8*
- En ajoutant une instruction de début (*begin* de numéro 0) et une instruction de fin (*end* de numéro 10), représenter ce problème sous forme d'un graphe orienté G.



- *Le sommet 0 doit précéder tous les autres sommets. Il suffit donc que 0 précède 1 et 5.*
- *Le sommet 10 doit être précédé par tous les autres sommets. Pour cela, il suffit que 9, 2 et 7 aient 10 comme successeur.*
- *Le graphe obtenu est un graphe orienté sans cycle.*

Q2. Par la suite, les successeurs d'un sommet seront traités par **ordre croissant de leurs numéros**.

- a. Indiquer l'ordre de visite des sommets pour un parcours en profondeur de G

0, 1, 2, 10, - 3, 6, 9 - 5, 4, 7, 8

- b. Indiquer l'ordre de visite des sommets pour un parcours en largeur de G

0, 1, 5, - 2, 3, - 4, 6, 8, - 7, 10, 9

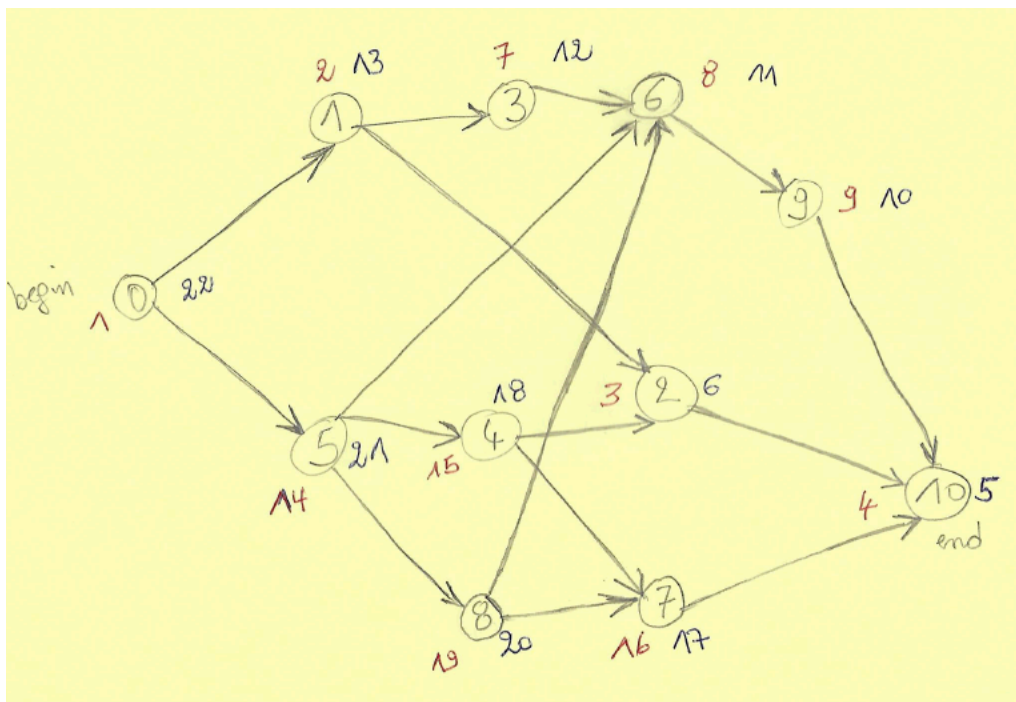
c. Effectuer un tri topologique du graphe G (en indiquant l'algorithme utilisé)

Rappel Algorithme #1 du 4.3 du CM 04

- On étiquette les sommets lors du parcours en profondeur en utilisant l'algorithme du 4.1
- On constate que pour tout arc (u, v) , on a : $FIN[v] < FIN[u]$
- Il suffit donc de trier les sommets par ordre de valeurs de FIN décroissantes.

Ci-dessous, l'application de cet algorithme à notre graphe.

- En rouge, et à gauche de chaque sommet i , sa date de début $DEBUT[i]$
- En bleu, et à droite de chaque sommet i , sa date de fin $FIN[i]$



Rq. Le graphe est parcouru en profondeur à partir du sommet 0.

Les sommets sont donc visités dans le même ordre que pour la réponse à la question Q2 (a).

On obtient un ordre topologique en classant les sommets par valeurs de FIN décroissantes

→ 0, 5, 8, 4, 7, 1, 3, 6, 9, 2, 10

Q3. En termes de graphes, à quoi correspond chaque solution de ce problème de génération de code ?

Toute solution correspond à un ordre topologique sur les sommets du graphe associé.