

Calcul de rang

Travail à effectuer en binômes.

Travail à effectuer

Vous devez mettre en œuvre un programme qui :

1. lit les arcs d'un graphe dans un fichier texte et en sauvegarde le contenu dans une structure de données de votre choix ;
2. calcule les rangs des sommets (et affiche le rang de chaque sommet).

Option : amélioration du programme de calcul de rang pour provoquer un abandon en cas d'existence de circuit.

Graphes que votre programme doit être capable de prendre en compte

Fichier de données

Langage de programmation

Idem TP 1

Rendu du travail

Vous devrez exécuter votre programme sur les graphes qui vous seront indiqués en séance de TP.

Vous devrez envoyer, à l'adresse qui vous sera communiquée en séance, un email avec les caractéristiques suivantes :

- Sujet : EFREI L3 TG TP2 – Nom1, Nom2
- Message : vide
- Pièces jointes : . Fichiers de code source + fichiers de données.
Code source uniquement, pas d'exécutable.
Pas de fichier compressé archive : mettez plusieurs pièces jointes si nécessaire.
. Traces d'exécution de votre programme dans un fichier « texte ».
Texte uniquement, en utilisant une police à taille fixe (par exemple Courier New).
Pas de copie d'écran en mode image.
Tous vos fichiers doivent être préfixés par vos deux noms.

Tout manquement à ces consignes sera pénalisé !

Rendu du travail en fin de séance de TP.

Éléments de notation

Ce TP sera a priori noté.

Seront pris en compte :

- Votre code lui-même : choix des structures de données (représentation du graphe et déroulement des algorithmes) et mise en œuvre de l'algorithme.
- Son adaptabilité à d'autres graphes que ceux demandés en traces d'exécution.
- Les commentaires, et plus généralement la clarté de votre code.
- Les traces fournies.

Toute récupération évidente, même partielle, de code fait par un autre groupe se verra sanctionnée par un 0/20 à tous les étudiants concernés.

Calcul de rang

Le rang des sommets est défini par :

$$\forall x \in S, \text{rang}(x) = \max_{y \in \Gamma^{-1}(x)} [\text{rang}(y)]$$

Le rang d'un sommet doit être plus grand que le rang de tous ses prédécesseurs.

Méthode de calcul :

Mise en œuvre récursive de la fonction « rang » :

Base : $\forall x \in S, \Gamma^{-1}(x) = \emptyset \Rightarrow \text{rang}(x) = 0$ (rang=0 pour les points d'entrée)

Induction : $\text{rang}(x) = \max_{y \in \Gamma^{-1}(x)} [\text{rang}(y)] + 1$

La fonction est appelée pour chaque point de sortie du graphe. Elle sauvegarde le rang de chaque sommet dans une structure de donnée de votre choix.

Pour sauvegarder les rangs calculés, on peut simplement utiliser un tableau `int * rangs`, de taille définie par le nombre de sommets. Les cases de ce tableau pourront être initialisées à '-1', valeur impossible pour un rang de sommet, qui sera interprétée comme « valeur pas encore définie ».

```
int fct_rang ( int S )

    si rang[S] ≠ -1 // rang déjà calculé
        alors retourner rang[S]

    si « S est un point d'entrée »
        alors rang[S] ← 0
        retourner 0

    //calcul en fonction des prédécesseurs
    int maxRangPred ← -1

    pour tout sommet X prédécesseur de S
    |   int rangPred ← fct_rang(X)
    |   si rangPred > maxRangPred
    |       alors maxRangPred ← rangPred

    rang[S] ← maxRangPred + 1
    retourner rang[S]
```