L3 Maths & Info

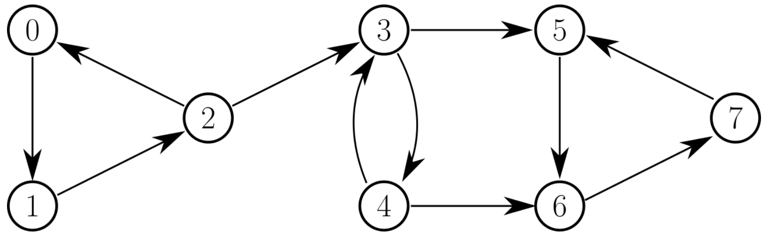
Cours d'Algorithmique 1

Devoir Final - Mercredi 16 décembre 2020 - 2h (9h - 11h)

Les notes de cours et de TD sont autorisées. Pour les questions demandant de dérouler un algorithme, vous veillerez à faire apparaître les étapes de calcul permettant de juger de votre bonne compréhension (le résultat final est rarement suffisant). Distinguez bien complexité temporelle et mémoire.

1 Algorithme de Kosaraju (10min)

Question 1. Appliquer l'algorithme de Kosaraju au graphe ci-dessous.



2 Graphe et degrés (40min)

Étant donné une liste de n entiers positifs d_1, d_2, ..., d_n, on souhaite déterminer efficacement s'il existe un graphe non orienté $G = (V, E)$ dont les sommets ont exactement les degrés donnés d_1, d_2, ..., d_n. C'est-à-dire, on peut écrire $V = \{v_1, ..., v_n\}$ tel que le degré de v_i soit d_i. Dans la suite, on notera bien v_i le sommet de degré d_i. On dit que $(d_1, ..., d_n)$ est une *séquence des degrés de* G. Ce graphe G ne doit pas contenir d'arête-boucle (arête dont les deux extrémités sont le même sommet), ou d'arcs multiples entre deux mêmes sommets.

Question 1. Donner un exemple de d_1, d_2, d_3, d_4 avec chaque $d_i < 3$ et $d_1 + d_2 + d_3 + d_4$ pair, mais pour lesquels il n'existe pas de graphe dont la séquence des degrés est (d_1, d_2, d_3, d_4).

Question 2. Supposons $d_1 \geq d_2 \geq ... \geq d_n$ et qu'il existe un graphe $G = (V, E)$ de séquence des degrés $(d_1, ..., d_n)$. On souhaite montrer qu'il existe un graphe qui a cette séquence des degrés, et dont les voisins de v_1 sont $v_2, v_3, ..., v_{1+d_1}$. L'idée est de transformer graduellement G en un graphe qui possède cette propriété additionnelle.

(a) Supposons que les voisins de v_1 ne sont pas $v_2, v_3, ..., v_{1+d_1}$. Montrer qu'il existe $i < j \leq n$ et $u \in V$ tels que $(v_1, v_i), (u, v_j) \notin E$ et $(v_1, v_j), (u, v_i) \in E$.

(b) Quels changements peut-on appliquer à G pour obtenir un graphe $G' = (V', E')$ avec la même séquence des degrés que G mais avec $(v_1, v_i) \in E'$?

(c) Montrer qu'il existe un graphe qui a la même séquence des degrés, et dont les voisins de v_1 sont $v_2, v_3, ..., v_{1+d_1}$.

Question 3. En utilisant les résultats de la question précédente, décrire un algorithme qui prend en entrée d_1, d_2, ..., d_n (pas forcément ordonnés), et retourne vrai s'il existe un graphe avec cette séquence des degrés. L'algorithme doit avoir une complexité polynomiale en n et en $m = \sum_{i=0}^{n} d_i$.

3 Sous-suite palindromique la plus longue (30min)

Un palindrome est une suite de caractères qui reste la même lorsque inversée, comme "non" ou "qazaq". La littérature regorge abondamment d'exemples de tels mots ou phrases : "À révéler mon nom, mon nom relèvera" (Cyrano de Bergerac).

Dans cet exercice, on se propose de rechercher dans une suite de caractères la longueur d'une plus grande sous-suite palindromique, c'est à dire la plus grande sous-suite constituant un palindrome. Par exemple un algorithme résolvant ce problème et prenant en entrée "bananepourantoine" devrait retourner 5 pour le palindrome "anana", obtenu comme suit : "bananepourantoine". Le problème d'optimisation associé est

- Entrée : une suite de caractères w ;
- Sortie : le plus grand k tel qu'il existe une sous-suite palindromique de taille k dans w.

Afin de simplifier le problème, nous considérons tous les caractères comme importants et différents (on n'identifie pas majuscule ni minuscule et les espaces ne sont pas sautés).

Question 1. Donner une plus grande sous-suite palindromique pour chacun des mots suivants : "annonerenchantant", "chercheunpalindrome" et "reverdetrouver".

Question 2. Donner le principe d'un algorithme résolvant le problème. Indiquer le paradigme de l'algorithme, la récurrence sur lequel il est basé. Prouver la relation de récurrence.

Question 3. Écrire le pseudo-code de l'algorithme. Commenter sa complexité et argumentez sa terminaison et sa correction.

4 NP-complétude (40min)

Le problème du couplage 3D est le suivant :

- Entrée : trois ensembles finis A, B, C de même cardinal n, disjoints deux à deux, et une relation $E \subseteq A \times B \times C$;
- Sortie : oui s'il existe $R \subseteq E$ tel que tout élément de $A \cup B \cup C$ apparaît dans exactement un triplet de R, non sinon.

La relation E est une relation de compatibilité : $(a, b, c) \in E$ s'appelle une *compatibilité* et signifie que les éléments a, b, et c peuvent être sélectionnées pour être ensembles. On pourrait imaginer un contexte heroic-fantasy où A est un ensemble d'armes, B de bracelets (magiques) et C des colliers mais que seules les combinaisons dans R sont compatibles.

Question 1. Donner une instance positive pour $n = 3$.

Nous nous proposons de donner plusieurs arguments pour démontrer l'appartenance du problème de couplage 3D à NP.

Question 2. Montrer que le problème du couplage 3D est dans NP.

Démontrons maintenant que le problème du couplage 3D est NP-dur. On rappelle que 3-SAT est le problème SAT où on restreint l'entrée aux formules sous forme normale conjonctive avec au plus 3 littéraux par clause. Aussi, introduisons la restriction 3-SATbis du problème 3-SAT, où chaque littéral apparaît au plus 2 fois dans la formule.

Question 3. Pour montrer 3-SATbis est NP-dur, faut-il faire une réduction de 3-SATbis à 3-SAT, ou de 3-SAT à 3-SATbis ? Expliquer informellement la réduction (5 lignes environ).

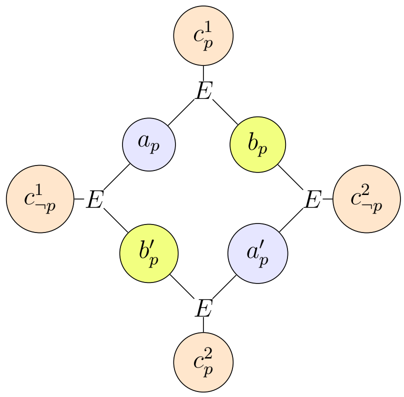
Maintenant attaquons nous à la démonstration de la NP-dureté du problème de couplage 3D. Pour cela on considère une instance ϕ de 3-SATbis, et on construit une instance $tr(\phi) = (A, B, C, E)$ du problème de couplage 3D comme suit.

Variables propositionnelles.

Pour chaque proposition atomique p apparaissant dans ϕ, nous introduisons les éléments a_p et a'_p dans A, les éléments b_p et b'_p dans B, et les éléments $c^1_{\neg p}$, $c^2_{\neg p}$, c^1_p, et c^2_p dans C. On introduit aussi les compatibilités suivantes :

$$(a_p, b'_p, c^1_{\neg p}), (a_p, b_p, c^1_p), (a'_p, b_p, c^2_{\neg p}), (a'_p, b'_p, c^2_p)$$

que l'on peut représenter graphiquement par



On interprète p vraie comme le fait que a_p, b'_p et $c^1_{\neg p}$ soient choisis ensemble. Les éléments $c^1_{\neg p}$ et $c^2_{\neg p}$ représentent les deux occurrences de $\neg p$ dans ϕ (s'il n'y en a qu'une alors $c^2_{\neg p}$ ne sert à rien !), les éléments c^1_p et c^2_p représentent les deux occurrences du littéral p dans ϕ (s'il n'y en a qu'une alors c^2_p ne sert à rien !).

Clauses. Pour une clause γ qui apparaît dans ϕ, on introduit un élément $a_\gamma \in A$ et un élément $b_\gamma \in B$. On ajoute des compatibilités entre a_γ, b_γ et les éléments de C qui représentent les occurrences de γ. Par exemple pour la clause $\gamma = (p \vee \neg q \vee r)$ en supposant qu'elle contienne la première occurrence de p, la deuxième occurrence de $\neg q$ et la première de r, on ajoute les compatibilités

$$(a_\gamma, b_\gamma, c^1_p), (a_\gamma, b_\gamma, c^2_{\neg q}), (a_\gamma, b_\gamma, c^1_r).$$

On note m le nombre de clauses et n le nombre de variables. Jusque là, A et B sont de cardinal $m + 2n$ et C de cardinal $4n$. On ajoute $2n - m$ couples arme-bracelet et qui sont compatibles avec tous les colliers.

Question 4. Montrer que si ϕ est satisfiable alors $tr(\phi)$ est positive (on demande un texte explicatif donnant tous les arguments, tout en étant léger sur la formalisation).

Question 5. Montrer que si $tr(\phi)$ est positive, alors ϕ est satisfiable (on demande un texte explicatif donnant tous les arguments, tout en étant léger sur la formalisation).