Révisions

SAT – st-connexité

Soit $G = (V, E)$ un graphe, et s, t deux sommets. On cherche à tester si s et t sont dans deux composantes connexes différentes.

1. Si uv est une arête de G, que dire des composantes connexes de u et v ?
2. Proposer une modélisation du problème en SAT.
3. Que pouvez-vous dire des clauses obtenues ? En déduire la complexité de l'algorithme issu de cette approche.

Partiel 2024 – Le pot de fin d'année

Une entreprise organise un pot pour la fin de l'année. L'entreprise est très hiérarchisée : à part le PDG, chaque employé a un unique supérieur direct, et un certain nombre de subordonnés directs. Le nouveau *happiness manager*, voulant promouvoir ses idées disruptives, déclare que pour éviter les tensions, il est impossible d'inviter à la fois une personne et son supérieur direct. Il est par contre acceptable d'inviter un employé et un de ses supérieurs hiérarchiques (tant que ce n'est pas son supérieur direct).

Le but de l'entreprise est de maximiser la *convivialité*, qui correspond au nombre de personnes invitées, d'après le *happiness manager*.

1. Résoudre le problème dans le cas où la PDG Alice a deux subordonnés directs Bob et Charline, Bob ayant deux subordonnés directs Daniel et Élise, tandis que Charline n'a que Françoise comme subordonnée.
2. Modéliser ce problème par un problème de graphes vu en cours/TD.
3. Quelle forme ont les graphes modélisant vos instances ?

On considère l'algorithme glouton suivant : si E est l'ensemble des employés, tant que E est non-vide, on invite une personne $p \in E$ la plus basse dans la hiérarchie (en cas d'égalité, on en choisit une arbitrairement), et on enlève p et son supérieur direct de E.

4. Montrer qu'il existe toujours une solution optimale où toutes les personnes les plus basses de la hiérarchie sont invitées.
5. En déduire que l'algorithme glouton renvoie une solution optimale.

Le *happiness manager* se rend compte que certaines personnes contribuent plus que d'autres à l'ambiance du pot lorsqu'elles sont invitées. Il attribue donc un score s_p à chaque personne p, et redéfinit la *convivialité* comme la somme des scores des personnes invitées.

6. Montrer que l'algorithme glouton précédent ne renvoie plus forcément une solution maximisant cette nouvelle notion de convivialité.

On numérote les employés de 1 à n dans l'ordre hiérarchique croissant, de sorte que chaque personne a un numéro plus grand que ses subordonnés (la PDG ayant le numéro n). On définit :

- C_i la convivialité maximale d'un pot où sont invités i et un sous-ensemble de ses subordonnés (directs ou indirects).
- D_i la convivialité maximale d'un pot où est invité un sous-ensemble des subordonnés (directs ou indirects) de i (mais pas i).

7. Donner une numérotation valide des employés de la question 1.
8. Que valent C_i et D_i si la i-ème personne est en bas de la hiérarchie (c'est-à-dire n'a pas de subordonné) ?

9. Exprimer C_i et D_i en fonction des C_j, D_j où j est un subordonné direct de i.
10. En déduire un algorithme de programmation dynamique résolvant le problème.
11. Quelle est sa complexité en temps et en espace ?

Examen 2023 – Tournois

On considère un tournoi joué par n équipes. L'équipe i rencontre l'équipe j exactement une fois et le match se termine soit par la victoire de l'équipe i soit par la victoire de l'équipe j. On peut donc représenter ce tournoi comme un graphe $G = (V, E)$ orienté où, pour chaque paire de sommets $(i, j) \in V^2$, on crée une arête orientée $i \to j$ de i vers j si i a battu j (et de j vers i autrement). On appelle *triangle orienté* un triplet de sommets (i, j, k) tels que $i \to j$, $j \to k$ et $k \to i$. (Un triangle est une "incohérence" pour le classement puisque cela signifie que l'on ne peut pas trouver d'ordre cohérent pour ces trois équipes).

1. Donner un tournoi à trois sommets sans triangle orienté. Quatre sommets ? n sommets ?
2. Montrer qu'il existe un unique tournoi à n sommets sans triangle orienté (à renommage près des sommets).
3. Donner un algorithme qui, étant donné un tournoi, compte le nombre total de triangles orientés. Autrement dit le nombre de triplets (i, j, k) qui forment des triangles orientés.
4. Les triangles d'un ensemble $\mathcal{S}$ de triangles orientés sont *sommets-disjoints* si pour toute paire de triangles orientés T_1, T_2 de $\mathcal{S}$, les sommets du triangle T_1 et du triangle T_2 sont deux à deux distincts. Modéliser avec un programme linéaire en nombres entiers (P) le problème qui consiste à trouver le nombre maximum de triangles orientés sommets-disjoints.
5. Un *feedback vertex set* est un ensemble de sommets X dont la suppression laisse un graphe sans triangle orienté. Autrement dit, il n'y a pas de triangle orienté dont les trois sommets sont dans $V \setminus X$.
   Modéliser avec un programme linéaire en nombres entiers (D) le problème qui consiste à trouver un feedback vertex set de taille minimum.
6. On suppose que les relaxations fractionnaires de (P) et (D) sont duales (ne le prouvez pas !). Que dire de la valeur optimale de (P) par rapport à celle de (D) ?
7. Montrer que l'ensemble des sommets apparaissant dans un ensemble maximal par inclusion de triangles orientés sommets-disjoints est un feedback vertex set de taille au plus trois fois la taille d'un feedback vertex set minimum.
8. En déduire un algorithme glouton qui donne un feedback vertex set de taille au plus trois fois l'optimum.