Faculté des Sciences Aix*Marseille Université

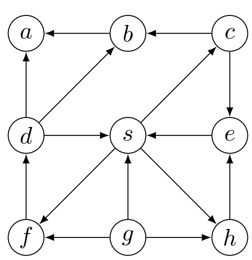
Année universitaire 2018/2019

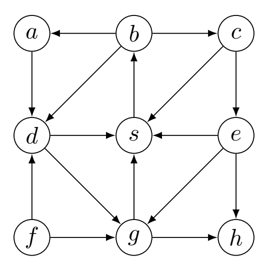
Site : ⊠ Luminy ⊠ St-Charles ☐ St-Jérôme ☐ Cht-Gombert ⊠ Aix-Montperrin ☐ Aubagne-SATIS
Sujet de : ⊠ 1$^{\text{er}}$ semestre ☐ 2$^{\text{ème}}$ semestre ☐ Session 2 Durée de l'épreuve : 2h
Examen de : L3 Nom du diplôme : Licence d'Informatique
Code du module : SIN5U03 Libellé du module : Algorithmique 2
Calculatrices autorisées : NON Documents autorisés : OUI, notes de Cours/TD/TP

Les questions sont pour la plupart indépendantes et ne sont pas classées par ordre de difficulté. Elles peuvent donc être traitées dans n'importe quel ordre. Une estimation de la difficulté de chaque question est donnée (⋆ facile, ⋆⋆ moyenne, ⋆⋆⋆ difficile), et pour certaines une borne maximum sur la longueur de la réponse, en nombre de caractères. Les réponses correctes dépassant excessivement *cette limite seront considérées fausses.*

1 Parcours de graphes

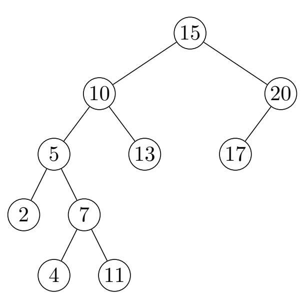
1. (⋆, 300 caractères) Quelle est la principale différence entre les parcours en largeur et en profondeur, en terme de structures de données utilisées pour les implémenter de manière non récursive ?
2. (⋆) Donner la complexité asymptotique de ces deux algorithmes.
3. (⋆, 200 caractères) Donner une application pour chacun de ces algorithmes.
4. (⋆) Effectuer un parcours en largeur dans le graphe de gauche et un parcours en profondeur dans le graphe de droite, en partant du sommet s : on traitera les arcs sortants dans l'ordre alphabétique croissant. Sur votre copie, dessiner les arborescences de parcours renvoyées par les algorithmes et, dans le cas du parcours en profondeur, donner les ordres d'entrée et de sortie (les dates de début et de fin par exemple).

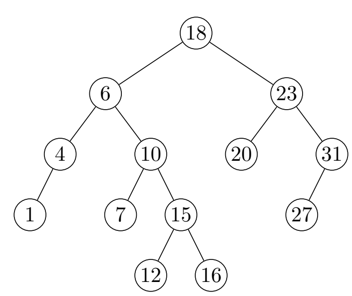




2 Algorithmes dans des arbres AVL

1. (⋆, 200 caractères) Quelle structure de données implémente les arbres binaires de recherche (et donc aussi les arbres AVL) ?
2. (⋆) Établir la liste de toutes les violations d'invariants d'arbres AVL dans l'arbre de gauche ci-dessous.
3. (⋆, 300 caractères) Pour quelle(s) raison(s) préfère-t-on utiliser des arbres AVL plutôt que des arbres binaires de recherche quelconques ?





4. ($\star\star$) À partir de l'arbre AVL de droite ci-dessus, insérer la clé 13 (en maintenant la propriété des arbres AVL).
5. ($\star\star$) À partir de l'arbre obtenu dans la question précédente, supprimer la clé 20 (en maintenant la propriété des arbres AVL).

3 Modélisation et flots

Vous êtes impliqué dans une expérience scientifique nécessitant des relevés de mesures atmosphériques à large échelle. L'expérience nécessite de connaître précisément un ensemble $P = \{p_1, p_2, \ldots, p_n\}$ de n propriétés atmosphériques (telles que le niveau d'ozone et de dioxyde de carbone à différents endroits). Vous avez à votre disposition un ensemble de m ballons que vous pouvez lancer pour faire ces mesures. Une fois lancé, chaque ballon peut mesurer seulement au plus deux propriétés. De plus, il y a des ballons de plusieurs types, chacun ne pouvant mesurer qu'un sous-ensemble des propriétés : pour chaque $i \in \{1, \ldots, m\}$, le ballon i ne peut mesurer que les propriétés $P_i \subseteq P$. Enfin, pour permettre d'obtenir des mesures plus fiables, on souhaite que chaque propriété p_i soit mesurée par au moins k ballons (évidemment, un ballon ne peut pas effectuer deux fois la même mesure).

Considérons un exemple avec $k = 2$, $n = 4$ propriétés p_1, p_2, p_3, p_4 et $m = 4$ ballons tels que les ballons 1 et 2 peuvent mesurer les propriétés $P_1 = P_2 = \{p_1, p_2, p_3\}$ et les ballons 3 et 4 peuvent mesurer les propriétés $P_3 = P_4 = \{p_1, p_3, p_4\}$. Une façon de faire en sorte que chaque propriété soit mesurée par au moins $k = 2$ ballons est la répartition suivante :

- le ballon 1 mesure les propriétés p_1 et p_2 ;
- le ballon 2 mesure les propriétés p_2 et p_3 ;
- le ballon 3 mesure les propriétés p_3 et p_4 ;
- le ballon 4 mesure les propriétés p_1 et p_4.

1. ($\star\star$) Proposer une modélisation du problème en terme de flot en précisant les sommets et les arêtes du réseau, ainsi que les capacités des arêtes. Illustrer la modélisation sur l'exemple précédent. Quel algorithme vu en cours permet alors de résoudre le problème en temps polynomial ?

Vous montrez la solution que vous avez obtenue à l'aide de votre algorithme à votre supérieur hiérarchique, mais il ne la trouve pas satisfaisante. En effet, le cahier des charges ne mentionnait pas une requête supplémentaire que la répartition des ballons doit satisfaire. Chaque ballon est construit par une des trois sociétés qui construisent de tels ballons atmosphériques. La requête supplémentaire à satisfaire est de faire en sorte qu'aucune propriété n'est mesurée uniquement (c'est-à-dire les k mesures) par des ballons du même constructeur.

Dans l'exemple précédent, supposons que le ballon 1 provienne du premier constructeur, les ballons 2 et 3 du second constructeur et le ballon 4 du troisième constructeur. Alors, la répartition proposée n'est plus correcte puisque la propriété p_3 n'est réalisée que par des ballons du second constructeur. Cependant, on pourrait utiliser les ballons 1 et 2 pour mesurer les propriétés p_1 et p_2, et les ballons 3 et 4 pour mesurer les propriétés p_3 et p_4.

2. ($\star\star\star$) Expliquer comment modifier votre modélisation afin de résoudre le problème avec cette nouvelle contrainte.

4 Formatage équilibré d'un paragraphe

Intéressons-nous au problème du formatage équilibré d'un paragraphe. On se donne ainsi une suite de n mots $m_1, \ldots, m_n$ de longueurs $\ell_1, \ell_2, \ldots, \ell_n$ et on souhaite imprimer ce paragraphe de manière équilibrée sur des lignes ne pouvant contenir qu'un maximum de M caractères chacune : on supposera donc que $\ell_i \leq M$ pour tout i. Le critère d'équilibre qu'on choisit est le suivant : si une ligne contient les mots de m_i à m_j inclus et qu'on laisse un caractère d'espacement entre chaque mot, le nombre de caractères d'espacements supplémentaires à la fin de la ligne est

$$f(i,j) = M - (j-i) - \sum_{k=i}^{j} \ell_k$$

L'objectif est de minimiser la somme des cubes des nombres de caractères d'espacement présents à la fin de chaque ligne, hormis la dernière ligne (qui peut donc être presque vide si nécessaire). On appelle *déséquilibre de la ligne* la quantité $f(i,j)^3$.

Par exemple, on a trois possibilités pour formater la première ligne de la Déclaration universelle des droits de l'homme, en lignes de 20 caractères (puisque le mot suivant est trop long pour pouvoir tenir sur celle-ci) :

```
Tous␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
les.................
....................
....................
```

```
Tous les␣␣␣␣␣␣␣␣␣␣␣␣
êtres...............
....................
....................
```

```
Tous les êtres␣␣␣␣␣␣
humains.............
....................
....................
```

Dans le premier cas, le déséquilibre de la première ligne est $f(1,1)^3 = (M - \ell_1)^3 = 16^3$; dans le second cas, il est égal à $f(1,2)^3 = (M - 1 - \ell_1 - \ell_2)^3 = 12^3$; dans le troisième cas, il est égal à $f(1,3)^3 = (M - 2 - \ell_1 - \ell_2 - \ell_3)^3 = 6^3$.

Un premier algorithme de formatage équilibré de paragraphe est la stratégie gloutonne consistant à remplir les lignes les unes après les autres en mettant à chaque fois le plus de mots possibles sur la ligne en cours.

1. ($\star$) Montrer sur un exemple simple que la stratégie gloutonne ne produit pas nécessairement le formatage optimal.

On applique désormais un algorithme de programmation dynamique. On note donc $d(j)$ la valeur minimale du déséquilibre total occasionné par le formatage du paragraphe des mots $m_1, \dots, m_j$.

2. ($\star\star$) Trouver une formule récursive pour calculer $d(j)$.
3. ($\star\star$) En déduire un algorithme de programmation dynamique calculant le déséquilibre minimal qu'on puisse obtenir pour une valeur de M et des longueurs $\ell_1, \dots, \ell_n$ données (qu'on suppose toutes inférieures à M). On supposera écrite une fonction calculant $f(i,j)$ pour un tableau de longueurs de mots et des indices i et j donnés.
4. ($\star$) Quelle est la complexité de l'algorithme précédent ?

5 Élimination de boules

On se donne une séquence de n boules colorées. On peut éliminer une boule de la séquence, au prix d'un euro. On peut aussi éliminer deux boules voisines gratuitement si elles ont la même couleur. L'objectif est de calculer le prix minimum à payer pour éliminer toutes les boules de la séquence.

Par exemple, en partant de la séquence de boules $RBBVRBVBR$ (rouge, bleu et vert), si on élimine la première boule verte, puis les deux boules rouges voisines, puis deux des trois boules bleues voisines, puis la boule verte, puis les deux boules bleues voisines, puis les deux boules rouges voisines, on a la suite d'élimination suivante :

$$RBBRVRBVBR \longrightarrow RBBRRBVBR \longrightarrow RBBBVBR \longrightarrow RBVBR \longrightarrow RBBR \longrightarrow RR \longrightarrow \varepsilon$$

Son prix total est de 2 euros.

Notons $c(i,j)$ le cout minimal d'élimination des boules comprises entre la boule d'indice i et la boule d'indice j (incluses). On admet en ce début d'exercice (cf question 5) que **si les boules i et j ont la même couleur, alors** $c(i,j) = c(i+1, j-1)$.

1. ($\star\star$) Donner une formule récursive permettant de calculer $c(i,j)$.
2. ($\star\star$) Écrire le pseudocode de l'algorithme pour résoudre le problème initial renvoyant le coût optimal d'élimination de la séquence de boules.
3. ($\star$) Quelle est sa complexité ?
4. ($\star$) Utiliser l'algorithme pour trouver l'optimum pour la séquence $RBBVRBVBR$.
5. ($\star\star\star$) Montrer la propriété de c admise en début d'exercice, à savoir que si les boules i et j ont la même couleur, alors $c(i,j) = c(i+1, j-1)$. *Indication : on pourra montrer qu'il existe toujours une stratégie d'élimination optimale des boules i à j qui rend voisines les boules i et j pour les éliminer gratuitement.*