Exercice 1

Montrer que dans tout graphe simple non orienté ayant au moins deux sommets, il existe toujours deux sommets de même degré.

Propriétés combinatoires des arbres

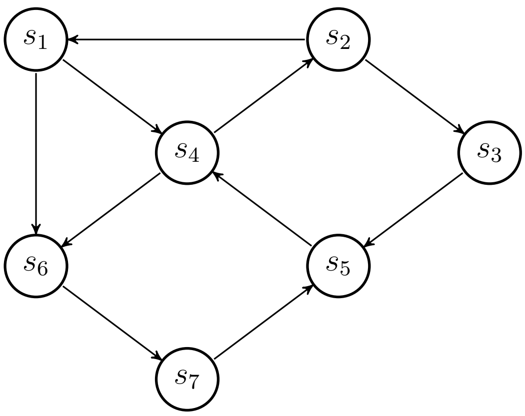
Exercice 2

1. Dessiner tous les arbres (non-isomorphes) à n sommets pour $n = 1, 2, 3, 4, 5, 6$.
2. Quel est le degré maximum d'un arbre à n sommets ? Justifier. Quelle est la structure des arbres qui, parmi tous les arbres à n sommets, ont le degré maximum le plus grand possible ? Pour une valeur de n donnée, un tel arbre est-il unique ?
3. Montrer que tout arbre à $n \geq 2$ sommets possède au moins deux sommets de degré 1. Quelle est la structure des arbres possédant exactement deux sommets de degré 1 ? Pour une valeur de $n \geq 2$ donnée, un tel arbre est-il unique ?
4. Soit G un arbre de degré maximum Δ. Montrer que G possède au moins Δ sommets de degré 1. Quelle est la structure des arbres qui, parmi tous les arbres de degré maximum Δ, possède un nombre minimum possible de sommets de degré 1 ? Pour une valeur de Δ donnée, un tel arbre est-il unique ?

Représentation de graphes orientés

Exercice 3

Donner les listes d'adjacence, la matrice d'adjacence et la matrice d'incidence du graphe orienté ci-dessous.



Exercice 4

Montrer que dans un graphe orienté, la somme des degrés entrants est égale à la somme des degrés sortants.

Calcul à partir de la matrice d'adjacence, de la matrice d'incidence, des listes d'adjacences

Exercice 5

Soit G un graphe (non-orienté) représenté par la matrice d'adjacence A.

Proposer un algorithme qui décide si G contient ou pas un sommet isolé (un sommet de degré 0).

(*) Proposer un algorithme qui calcule le degré maximum de G.

(*) Proposer un algorithme qui calcule le degré minimum de G.

Exercice 6

Soit G un graphe (non-orienté) représenté par la matrice d'incidence B.

Proposer un algorithme qui calcule le degré maximum de G.

(*) Proposer un algorithme qui décide si G contient ou pas un sommet isolé.

(*) Proposer un algorithme qui calcule le degré minimum de G.

Exercice 7

Soit G un graphe (non-orienté) représenté par les listes d'adjacence $Adj[\,]$.

Proposer un algorithme qui calcule le degré minimum de G.

(*) Proposer un algorithme qui décide si G contient ou pas un sommet isolé.

(*) Proposer un algorithme qui calcule le degré maximum de G.

Notation $O()$, notion de complexité

Exercice 8

Soit $f(n) = \frac{1}{3}n\left(n + \frac{1}{2}\right)(n+1)$. Montrer que $f(n) = \mathcal{O}(n^3)$

Les relations suivantes sont elles vraies ?

i. $f(n) = \mathcal{O}(n^{10})$

ii. $f(n) = \frac{1}{3}n^3 + \mathcal{O}(n^2)$

iii. $\mathcal{O}(n^2) = \mathcal{O}(n^3)$

iv. $\mathcal{O}(n^3) = \mathcal{O}(n^2)$

v. $\frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3)$

Exercice 9

Étudier la complexité des algorithmes conçus dans les exercices 5 à 7.

Exercice 10

On se donne un graphe orienté à n sommets et m arcs. On suppose que le degré (nombre d'arcs entrant ou sortant) de chaque sommet est au plus d.

Donner l'algorithme et une majoration (la plus basse possible en fonction de n, m, d) pour la complexité de chacune des opérations suivantes :

1. faire afficher tous les successeurs d'un sommet s donné
2. faire afficher tous les prédécesseurs d'un sommet s donné
3. déterminer si un graphe est sans boucle (une boucle est un arc $s \to s$)

suivant qu'on utilise, pour représenter le graphe, une matrice d'adjacence `A[i,j]`, une matrice d'incidence `B[u,e]`, ou des listes de successeurs `Adj[u]`.