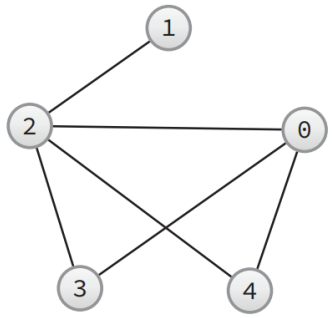
DS 2
Option informatique MP

Exercice E3A 2018



1. Écrire la matrice d'adjacence du graphe ci-dessus.
2. Écrire en Caml une fonction `chemin : int array array -> int list -> bool` qui prend en entrée la matrice d'adjacence d'un graphe et un chemin (une liste de sommets du graphe) et qui vérifie si ce chemin est possible dans le graphe. Par exemple, sur le graphe ci-dessus, avec le chemin `[2;1;0;4]` la fonction `chemin` doit renvoyer `false` car les sommets `1` et `0` ne sont pas connectés. Avec le chemin `[1;2;3]` la fonction `chemin` doit renvoyer `true` car les sommets `1` et `2` sont connectés, ainsi que les sommets `2` et `3`.
3. Écrire une fonction Caml `atteignable : int list array -> int -> int -> bool` telle que, si `g` est un graphe orienté représenté par liste d'adjacence et `r`, `u` deux sommets de `g`, `atteignable g r u` renvoie `true` si et seulement si il existe un chemin de `r` à `u` dans `g`.

Extrait E3A 2019

Diamètre d'un graphe

Dans cet exercice, on considère des graphes non orientés connexes. Les sommets d'un graphe à n sommets ($n \in \mathbb{N}^*$) sont numérotés de 0 à $n-1$. On suppose qu'aucune arête ne boucle sur un même sommet.

Un *chemin de longueur* $p \in \mathbb{N}$ d'un sommet a vers un sommet b dans un graphe est la donnée de $p+1$ sommets $s_0, s_1, \ldots, s_p$ tels que $s_0 = a$, $s_p = b$ et, pour tout $1 \leqslant k \leqslant p$, les sommets s_{k-1} et s_k sont reliés par une arête.

Un *plus court chemin* d'un sommet a vers un sommet b dans un graphe G est un chemin de longueur minimale parmi tous les chemins de a vers b. Sa longueur est notée $d_{\mathrm{G}}(a, b)$.

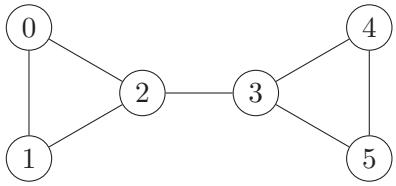
Le *diamètre* d'un graphe G, noté diam(G), vaut le maximum des longueurs des plus courts chemins entre deux sommets du graphe G. Autrement dit,

$$\mathrm{diam(G)} = \max_{a,b \text{ sommets de G}} d_{\mathrm{G}}(a, b)$$

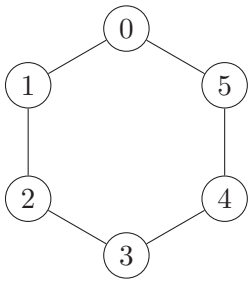
Un *chemin maximal* d'un graphe G est un plus court chemin de G de longueur diam(G).

Partie 1 – Exemples de graphes

15 Donner sans justification le diamètre et les chemins maximaux pour chacun des deux graphes G_1 et G_2 ci-dessous.



graphe G_1



graphe G_2

En OCaml, les graphes sont représentés par liste d'adjacence et implémentés par le type

```
type graphe = int list array;;
```

16 Graphes de diamètre maximal.

16a) Dessiner sans justification un graphe à 5 sommets ayant un diamètre le plus grand possible.

16b) Écrire en OCaml une fonction `diam_max` de type `int -> graphe` qui prend en argument un entier naturel n non nul et qui renvoie un graphe à n sommets de diamètre maximal.

17 Graphes de diamètre minimal.

17a) Dessiner sans justification un graphe à 5 sommets ayant un diamètre le plus petit possible.

17b) Écrire en OCaml une fonction `diam_min` de type `int -> graphe` qui prend en argument un entier naturel n non nul et qui renvoie un graphe à n sommets de diamètre minimal.

Tournez la page S.V.P.

Partie 2 – Algorithmes de calcul du diamètre

Dans cette partie, on suppose que les graphes sont représentés par listes d'adjacence.

18 Donner l'entrée et la sortie de l'algorithme de Dijkstra. Comment cet algorithme permet-il de calculer le diamètre d'un graphe ?

19 Quel parcours de graphe peut être utilisé pour le calcul du diamètre ?

20 Laquelle des deux méthodes précédentes est la mieux adaptée pour calculer le diamètre d'un graphe ?

Partie 3 – Diamètre d'un arbre binaire

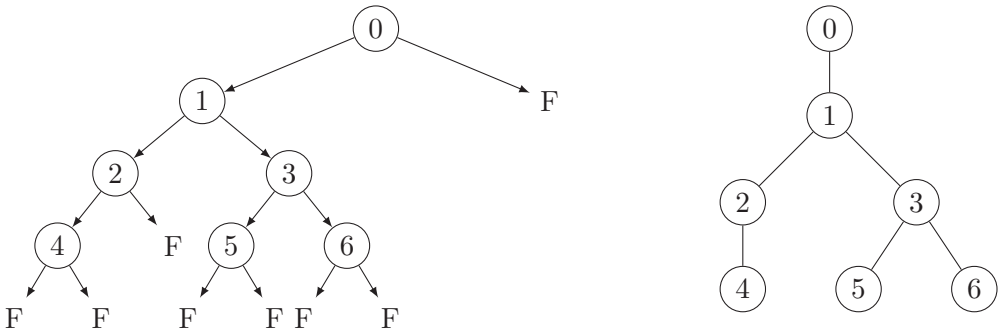
Dans cette partie, on s'intéresse aux arbres binaires, qui sont des cas particuliers de graphes. On travaille avec une représentation spécifique de ces graphes particuliers, implémentée en OCaml par le type suivant :

```
type arbre = Feuille | Noeud of int * arbre * arbre ;;
```

Le *graphe sous-jacent* $G_{\mathcal{A}}$ à un arbre binaire $\mathcal{A}$ est défini comme le graphe orienté dont

- les sommets correspondent aux nœuds de l'arbre (et pas aux feuilles) ;
- les arêtes correspondent aux branches de l'arbre reliant deux nœuds (et non pas celles reliant un nœud à une feuille).

Le diamètre d'un arbre binaire est alors défini comme le diamètre de son graphe sous-jacent.
Voici un exemple d'arbre binaire $\mathcal{A}$ (à gauche) et de son graphe sous-jacent $G_{\mathcal{A}}$ (à droite) :



21 Donner l'expression OCaml représentant l'arbre $\mathcal{A}$ de l'exemple. Donner le diamètre de $\mathcal{A}$ et les chemins maximaux du graphe sous-jacent $G_{\mathcal{A}}$.

22 Quel est le nombre r d'arêtes du graphe sous-jacent à un arbre binaire possédant n nœuds ?

Une première approche pour calculer le diamètre d'un arbre consiste à le transformer en un graphe et à employer un algorithme général sur les graphes de la partie 2.

23 Écrire en OCaml une fonction `nb_noeuds` de type `arbre -> int` qui renvoie le nombre de nœuds d'un arbre binaire donné en argument.

24 Écrire en OCaml une fonction `numerotation` de type `arbre -> arbre` qui prend en argument un arbre binaire $\mathcal{A}$ à n nœuds et qui renvoie un arbre binaire $\mathcal{A}'$ de même graphe sous-jacent que $\mathcal{A}$ et dont les nœuds sont étiquetés de 0 à $n-1$.

25 Écrire en OCaml une fonction `arbre_vers_graphe` de type `arbre -> graphe` qui prend en argument un arbre binaire $\mathcal{A}$ à n nœuds étiquetés de 0 à $n-1$ et qui renvoie le graphe $\mathrm{G}_{\mathcal{A}}$ sous-jacent à $\mathcal{A}$ (le type `graphe` est défini dans la partie 1).

26 Décrire un algorithme qui calcule le diamètre d'un arbre de type `arbre` en se ramenant à un graphe. Quelle est sa complexité ?

Une seconde approche pour calculer le diamètre d'un arbre consiste à employer une technique diviser-pour-régner. Pour tout arbre $\mathcal{A}$ non réduit à une feuille, de la forme `Noeud (x, arbre_g, arbre_d)`, on note

- $\mathcal{A}_g$ le fils gauche de $\mathcal{A}$, représenté par `arbre_g` ;
- $\mathcal{A}_d$ le fils droit de $\mathcal{A}$, représenté par `arbre_d`.

La hauteur de l'arbre $\mathcal{A}$, notée $h(\mathcal{A})$, est la longueur du plus long chemin descendant de la racine vers une feuille. Dans l'arbre exemple de la partie 3, l'arbre $\mathcal{A}$ est de hauteur 4.

27 Quelle est la longueur d'un chemin maximal passant par la racine ?

28 Écrire en OCaml une fonction `diam_arbre` de type `arbre -> int` qui calcule le diamètre d'un arbre donné en argument. Cette fonction devra être de complexité linéaire en le nombre de nœuds de l'arbre.