

Arbres binaires

Dans cette section les arbres sont définis par le type polymorphe :

```
type 'a arbre = Vide | Noeud of 'a arbre * 'a * 'a arbre
```

- 1) Une feuille est un nœud dont les deux fils sont vides. Écrire une fonction qui compte le nombre de feuilles d'un arbre : `nb_feuilles : 'a arbre -> int`

Quelle est la complexité de cette fonction ?

- 2) Rappeler la définition de la profondeur (ou hauteur) d'un nœud, ainsi que la taille d'un arbre binaire.

Écrire les fonctions : `profondeur : 'a arbre -> int` et `taille : 'a arbre -> int`.

Donner les complexités de ces fonctions ?

- 3) Quelle relation d'égalité y a-t-il entre le nombre de nœuds vides d'un arbre binaire et le nombre de nœuds non vides ?

Quelle relation d'inégalité permet de majorer la taille en fonction de la profondeur ?

- 4) Rappeler la définition d'un tas-max.

Écrire en pseudo-code l'algorithme de suppression du maximum d'un tas-max (le tas étant implémenté dans un arbre binaire).

Quelle est la complexité de cet algorithme, en fonction du nombre d'éléments présents dans le tas ?

- 5) Écrire en Caml une fonction `ajoute : int array -> int -> unit` qui permet d'ajouter un élément dans un tas-max implémenté dans un tableau. On prendra comme convention que la taille du tas est stockée dans la case d'indice 0, les données commençant à la case d'indice 1.

Quelle est la complexité de votre fonction, en fonction du nombre d'éléments présents dans le tas ?

- 6) Rappeler la définition d'un arbre binaire de recherche.

- 7) Écrire une fonction de type `int -> int arbre -> bool` qui permet de tester si un élément appartient ou non à un arbre binaire de recherche.

Quelle est sa complexité ?

- 8) Rappeler le principe de l'algorithme d'insertion d'un élément dans un arbre binaire de recherche. Écrire la fonction `insert : int -> int arbre -> int arbre`.

- 9) Rappeler l'algorithme de suppression de la racine dans un arbre binaire de recherche. Rappeler le principe de suppression d'un élément dans un arbre binaire de recherche. S'il reste du temps, écrire ces fonctions.

Logique

10) Le type suivant permet de coder des formules logiques :

```
type formule = Variable of char
              | Constante of bool
              | Non of formule
              | Ou of formule * formule
              | Et of formule * formule
```

Pour les instances de type `formule`, expliquer la différence entre les notions d'égalité suivantes :

- égalité syntaxique ;
- égalité sémantique ;

11) Qu'est-ce qu'une tautologie ? Une antilogie ? Une formule satisfiable ?

Graphes

12) Définir les notions suivantes : degré d'un sommet dans un graphe orienté ; composante connexe dans un graphe non orienté.

13) Programmer en Caml une fonction permettant grâce à un parcours en profondeur de connaître les sommets qui sont dans la même composante connexe qu'un sommet donné, dans un graphe non orienté codé par listes d'adjacence.

Quelle est la complexité de votre fonction ?

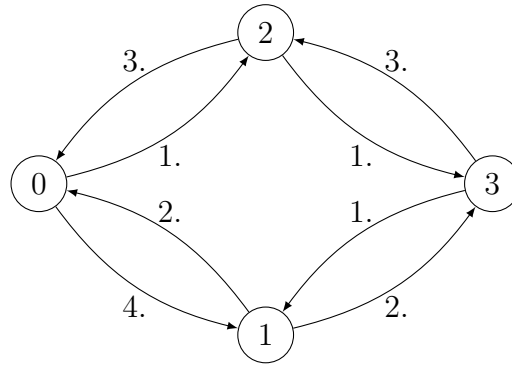
14) Même question mais avec un parcours en largeur. On pourra supposer qu'on dispose d'une structure de file dans laquelle les opérations élémentaires se font en temps constant (création d'une file vide, tester si la file est vide, enfiler un nouvel élément, défiler l'élément le plus ancien).

15) Programmer en pseudo-langage l'algorithme de Floyd-Warshall dans un graphe orienté pondéré codé par matrice d'adjacence.

Quelle est la complexité de votre fonction ?

Est-il possible d'exécuter l'algorithme de Floyd-Warshall lorsque certains poids sont négatifs ?

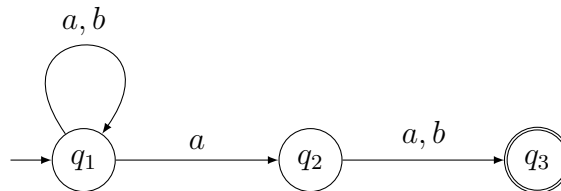
16) On considère le graphe orienté pondéré suivant :



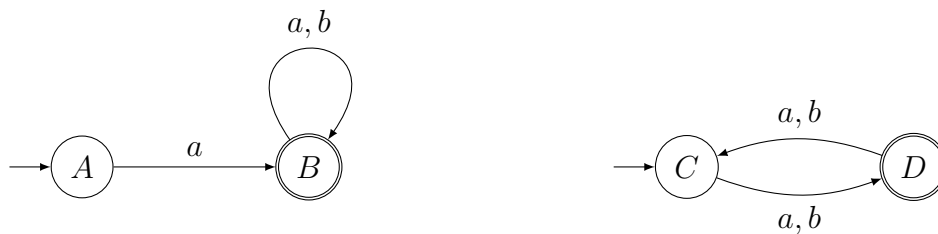
Détailler à la main les différentes étapes du parcours de ce graphe en partant du sommet 0 et en appliquant l'algorithme de Dijkstra.

Langages et automates

- 17) Rappeler la définition d'un automate. Rappeler la définition d'un automate déterministe, d'un automate complet.
- 18) Comment rendre complet un automate ?
- 19) Montrer que le langage $L = \{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas rationnel.
- 20) Rappeler l'algorithme de détermination d'un automate. Cet algorithme donne-t-il un automate complet ? Déterminer l'automate :



- 21) Rappeler la construction d'un automate produit de deux automates. Quel langage est reconnu par l'automate produit ? Faire cette construction dans le cas des automates :



Quels sont les langages reconnus par ces différents automates ?

22) Glushkov

Soit e l'expression rationnelle $(a + b)^*a$. On cherche à construire un automate reconnaissant le langage défini par e avec l'algorithme de Glushkov. Effectuer successivement les opérations suivantes :

- linéarisation par marquage ;
- description locale du langage ;
- construction de l'automate local ;
- suppression des marques ;
- (eventuellement) déterminisation de l'automate obtenu.