

Terminale NSI Evaluation n°2 S1

Nom :

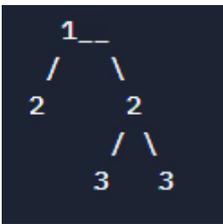
Prénom :

Note :

Exercice n° 1 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres droit et gauche de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la taille d'un arbre en fonction des tailles de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et taille**.
4. On nomme T_n la taille de l'arbre de Fibonacci A_n .

A l'aide de votre programme Python, donner les valeurs de $T(n)$ pour des valeurs de n comprises entre 0 et 5.

Conjecturer une relation de récurrence entre T_n , T_{n-1} et T_{n-2} .

Exercice n° 2 -- 2 pts

On souhaite insérer dans une structure de données les différentes hauteurs d'un arbre au fil des années sans y insérer les années. La structure de données la plus adaptée en Python est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Exercice n° 3 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

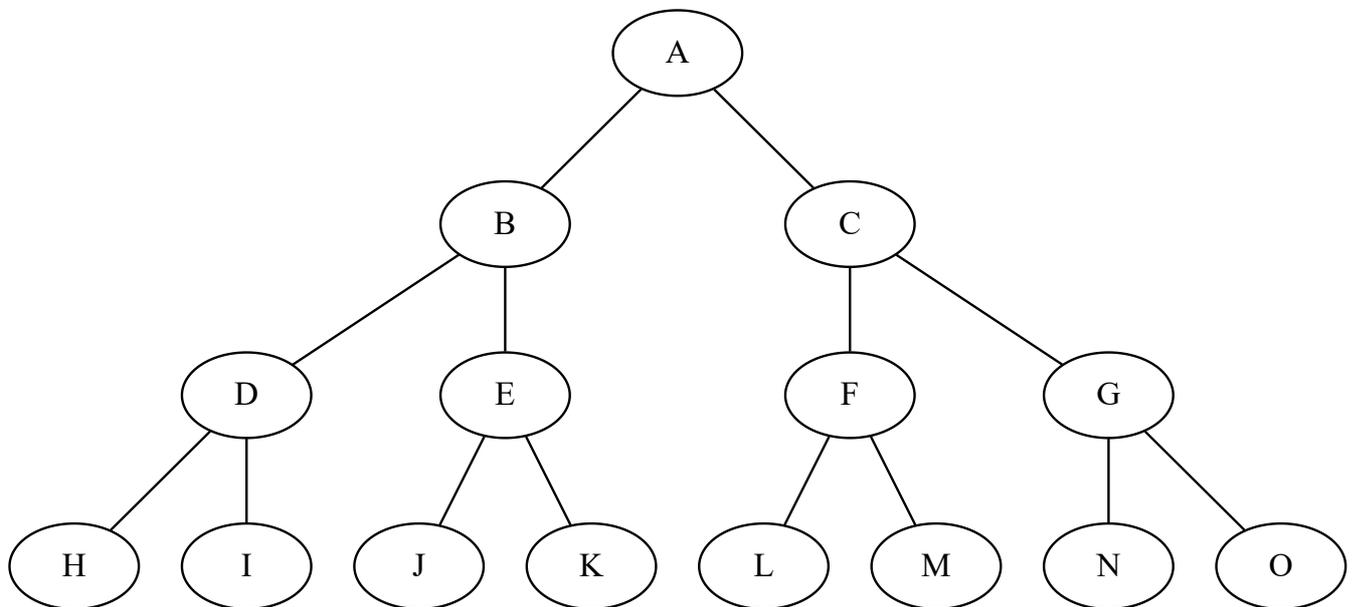
1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Exercice n° 4 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de la **hauteur** d'un arbre.
3. Donner la définition d'un arbre **dégénéré**.

On considère l'arbre suivant :



4. Donner les caractéristiques de cet arbre.
5. Cet arbre est-il un arbre binaire complet ? Justifier.
6. Calculer (en détaillant) les mesures suivantes de cet arbre :
 - Sa taille
 - Sa hauteur
 - Sa longueur de cheminement : LC
 - Sa longueur de cheminement externe : LCE
 - Sa longueur de cheminement interne : LCI
 - Sa profondeur moyenne : PM
 - Sa profondeur moyenne externe : PME
 - Sa profondeur moyenne interne : PMI

Ranger par ordre croissant les 3 profondeurs calculées.

Exercice n° 5 -- 3 pts

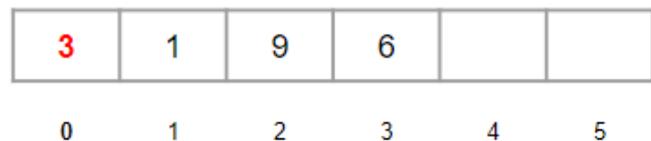
Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



Ecrire en Python (ou en pseudo-code) le code de la fonction `RECHERCHER(L, e)` qui recherche l'élément e de la liste L et retourne son index si l'élément est trouvé et `None` sinon.

Exercice n° 6 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 1, 2 et 3.
3. Où se trouve le minimum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste `[racine(A), SAG(A), SAD(A)]` où :

- `racine(A)` est la fonction qui retourne la racine de l'arbre A ;
- `SAG(A)` est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- `SAD(A)` est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python `minimum(A)` qui retourne le minimum de l'arbre binaire de recherche A .

Exercice n° 7 -- 3 pts

1. Sur quel principe sont basées les **files** ?
2. Donner les opérations de base que l'on peut réaliser sur une **file**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Terminale NSI Evaluation n°2 S2

Nom :

Prénom :

Note :

Exercice n° 1 -- 3 pts

1. Sur quel principe sont basées les **pires** ?
2. Donner les opérations de base que l'on peut réaliser sur une **pile**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Exercice n° 2 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 4, 5 et 6.
3. Où se trouve le maximum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste $[racine(A), SAG(A), SAD(A)]$ où :

- $racine(A)$ est la fonction qui retourne la racine de l'arbre A ;
- $SAG(A)$ est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- $SAD(A)$ est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python $maximum(A)$ qui retourne le maximum de l'arbre binaire de recherche A .

Exercice n° 3 -- 2 pts

On souhaite insérer dans une structure de données les différentes hauteurs d'un arbre au fil des années sans y insérer les années. La structure de données la plus adaptée en Python est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Exercice n° 4 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

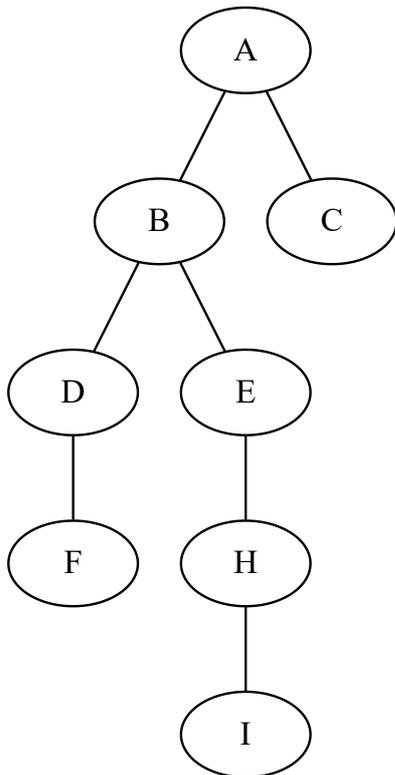
1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Exercice n° 5 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de la **taille** d'un arbre.
3. Donner la définition d'un arbre **équilibré**.

On considère l'arbre suivant :

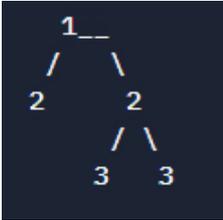


4. Donner les caractéristiques de cet arbre.
5. Cet arbre est-il un arbre binaire ? Justifier.
6. Calculer (en détaillant) les mesures suivantes de cet arbre :
 - Sa taille
 - Sa hauteur
 - Sa longueur de cheminement
 - Sa longueur de cheminement externe
 - Sa longueur de cheminement interne
 - Sa profondeur moyenne
 - Sa profondeur moyenne externe
 - Sa profondeur moyenne interne

Exercice n° 6 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres droit et gauche de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la taille d'un arbre en fonction des tailles de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et taille**.
4. On nomme T_n la taille de l'arbre de Fibonacci A_n .

A l'aide de votre programme Python, donner les valeurs de $T(n)$ pour des valeurs de n comprises entre 0 et 5.

Conjecturer une relation de récurrence entre T_n , T_{n-1} et T_{n-2} .

Exercice n° 7 -- 3 pts

Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



3	1	9	6		
0	1	2	3	4	5

Ecrire en Python (ou en pseudo-code) le code de la fonction `SUPPRIMER(L, i)` qui supprime l'élément à la position i de la liste L .

Terminale NSI Evaluation n°2 S3

Nom :

Prénom :

Note :

Exercice n° 1 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 1, 2 et 3.
3. Où se trouve le minimum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste $[racine(A), SAG(A), SAD(A)]$ où :

- $racine(A)$ est la fonction qui retourne la racine de l'arbre A ;
- $SAG(A)$ est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- $SAD(A)$ est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python $minimum(A)$ qui retourne le minimum de l'arbre binaire de recherche A .

Exercice n° 2 -- 3 pts

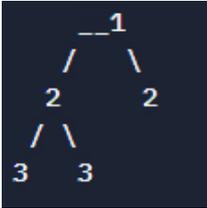
1. Sur quel principe sont basées les **pires** ?
2. Donner les opérations de base que l'on peut réaliser sur une **pile**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Exercice n° 3 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres gauche et droit de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la hauteur d'un arbre en fonction des hauteurs de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et hauteur**
4. Rappeler la définition d'un arbre équilibré.

Les arbres de Fibonacci sont-ils équilibrés ? Justifier.

Exercice n° 4 -- 3 pts

Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



3	1	9	6		
0	1	2	3	4	5

Ecrire en Python (ou en pseudo-code) le code de la fonction `SUPPRIMER(L, i)` qui supprime l'élément à la position i de la liste L .

Exercice n° 5 -- 2 pts

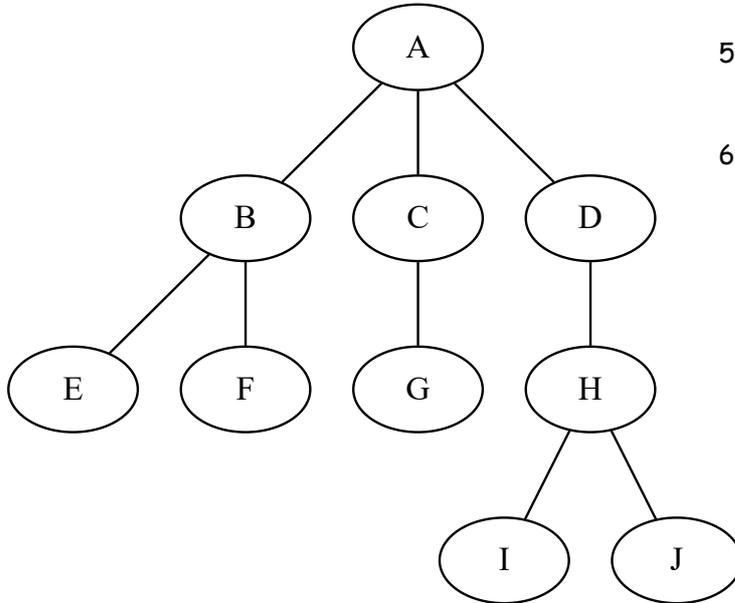
Dans une association, on souhaite enregistrer une liste d'informations pour chaque adhérent : taille, poids, etc. La meilleure structure de données pour cela est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Exercice n° 6 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de l'**arité** d'un arbre.
3. Donner la définition de la **hauteur** d'un arbre.

On considère l'arbre suivant :



4. Donner les caractéristiques de cet arbre.

5. Cet arbre est-il un arbre binaire complet ? Justifier.

6. Calculer (en détaillant) les mesures suivantes de cet arbre :

- Sa taille
- Sa hauteur
- Sa longueur de cheminement
- Sa longueur de cheminement externe
- Sa longueur de cheminement interne
- Sa profondeur moyenne
- Sa profondeur moyenne externe
- Sa profondeur moyenne interne

Ranger par ordre croissant les 3 profondeurs calculées.

Exercice n° 7 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Terminale NSI Evaluation n°2 S4

Nom :

Prénom :

Note :

Exercice n° 1 -- 3 pts

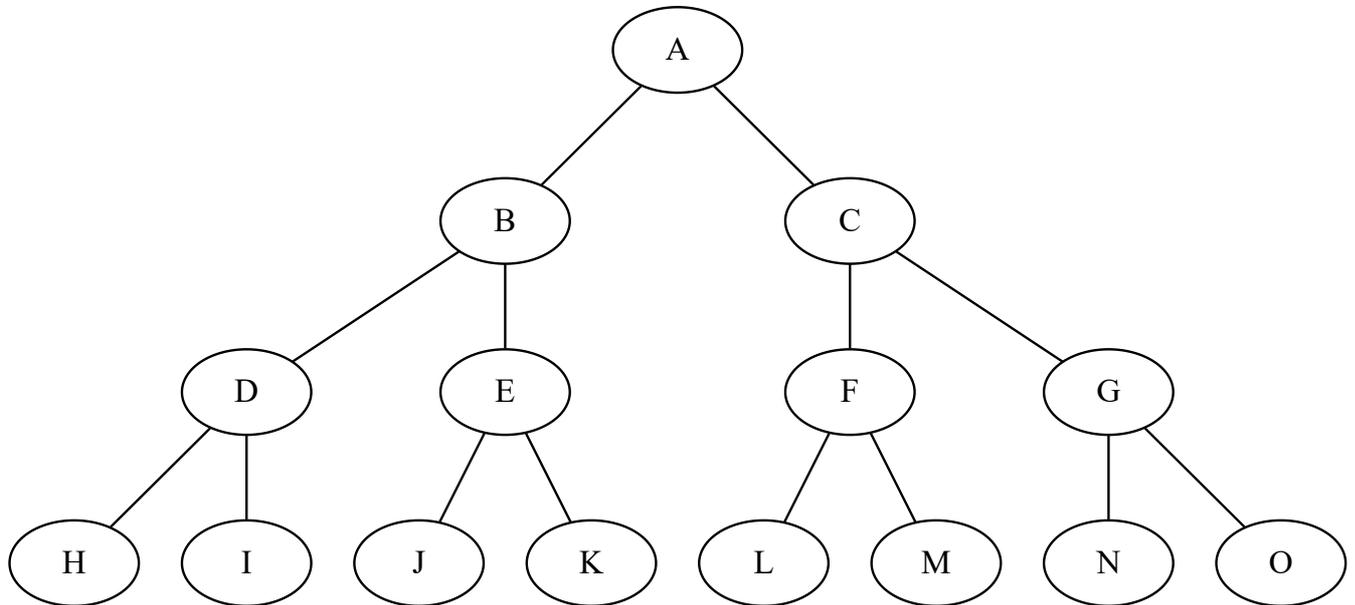
1. Sur quel principe sont basées les **files** ?
2. Donner les opérations de base que l'on peut réaliser sur une **file**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Exercice n° 2 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de la **hauteur** d'un arbre.
3. Donner la définition d'un arbre **dégénéré**.

On considère l'arbre suivant :



4. Donner les caractéristiques de cet arbre.
5. Cet arbre est-il un arbre binaire complet ? Justifier.
6. Calculer (en détaillant) les mesures suivantes de cet arbre :
 - Sa taille
 - Sa hauteur
 - Sa longueur de cheminement : LC
 - Sa longueur de cheminement externe : LCE
 - Sa longueur de cheminement interne : LCI
 - Sa profondeur moyenne : PM
 - Sa profondeur moyenne externe : PME
 - Sa profondeur moyenne interne : PMI

Ranger par ordre croissant les 3 profondeurs calculées.

Exercice n° 3 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

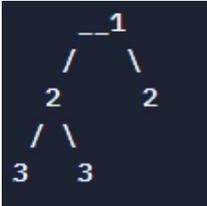
1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Exercice n° 4 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres gauche et droit de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la hauteur d'un arbre en fonction des hauteurs de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et hauteur**
4. Rappeler la définition d'un arbre équilibré.

Les arbres de Fibonacci sont-ils équilibrés ? Justifier.

Exercice n° 5 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 1, 2 et 3.
3. Où se trouve le minimum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste $[racine(A), SAG(A), SAD(A)]$ où :

- $racine(A)$ est la fonction qui retourne la racine de l'arbre A ;
- $SAG(A)$ est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- $SAD(A)$ est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python $minimum(A)$ qui retourne le minimum de l'arbre binaire de recherche A .

Exercice n° 6 -- 3 pts

Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



3	1	9	6		
0	1	2	3	4	5

Ecrire en Python (ou en pseudo-code) le code de la fonction *INSERER*(L, e, i) qui insère l'élément e à la position i de la liste L .

Exercice n° 7 -- 2 pts

On souhaite insérer dans une structure de données les différentes hauteurs d'un arbre au fil des années sans y insérer les années. La structure de données la plus adaptée en Python est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Terminale NSI Evaluation n°2 S5

Nom :

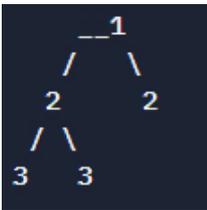
Prénom :

Note :

Exercice n° 1 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres gauche et droit de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



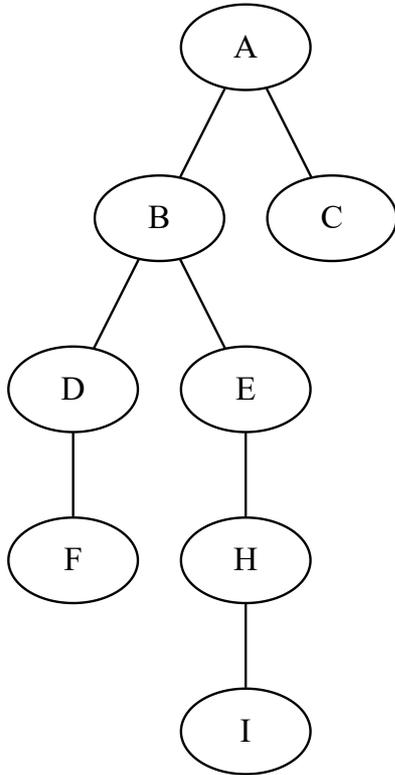
1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la hauteur d'un arbre en fonction des hauteurs de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et hauteur**
4. Rappeler la définition d'un arbre équilibré.

Les arbres de Fibonacci sont-ils équilibrés ? Justifier.

Exercice n° 2 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de la **taille** d'un arbre.
3. Donner la définition d'un arbre **équilibré**.

On considère l'arbre suivant :



4. Donner les caractéristiques de cet arbre.
5. Cet arbre est-il un arbre binaire ? Justifier.
6. Calculer (en détaillant) les mesures suivantes de cet arbre :
 - Sa taille
 - Sa hauteur
 - Sa longueur de cheminement
 - Sa longueur de cheminement externe
 - Sa longueur de cheminement interne
 - Sa profondeur moyenne
 - Sa profondeur moyenne externe
 - Sa profondeur moyenne interne

Exercice n° 3 -- 2 pts

Dans une association, on souhaite enregistrer une liste d'informations pour chaque adhérent : taille, poids, etc. La meilleure structure de données pour cela est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Exercice n° 4 -- 3 pts

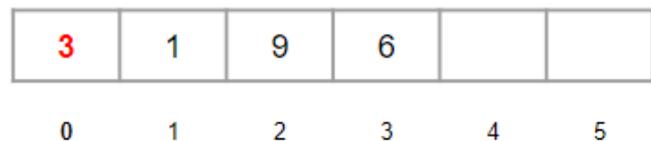
Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



Ecrire en Python (ou en pseudo-code) le code de la fonction $INSERER(L, e, i)$ qui insère l'élément e à la position i de la liste L .

Exercice n° 5 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 1, 2 et 3.
3. Où se trouve le minimum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste $[racine(A), SAG(A), SAD(A)]$ où :

- $racine(A)$ est la fonction qui retourne la racine de l'arbre A ;
- $SAG(A)$ est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- $SAD(A)$ est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python $minimum(A)$ qui retourne le minimum de l'arbre binaire de recherche A .

Exercice n° 6 -- 3 pts

1. Sur quel principe sont basées les **files** ?
2. Donner les opérations de base que l'on peut réaliser sur une **file**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Exercice n° 7 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Terminale NSI Evaluation n°2 S6

Nom :

Prénom :

Note :

Exercice n° 1 -- 3 pts

1. Donner la définition d'un arbre binaire de recherche (ABR).
2. Dessiner tous les arbres binaires de recherche formés des noeuds 4, 5 et 6.
3. Où se trouve le maximum d'un arbre binaire de recherche

On dispose d'un arbre binaire de recherche A représenté par la liste $[racine(A), SAG(A), SAD(A)]$ où :

- $racine(A)$ est la fonction qui retourne la racine de l'arbre A ;
- $SAG(A)$ est la fonction qui retourne le sous arbre gauche de l'arbre A ;
- $SAD(A)$ est la fonction qui retourne le sous arbre droit de l'arbre A .

Ecrire sur votre feuille la fonction récursive Python $maximum(A)$ qui retourne le maximum de l'arbre binaire de recherche A .

Exercice n° 2 -- 2 pts

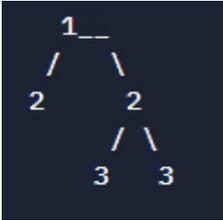
On souhaite insérer dans une structure de données les différentes hauteurs d'un arbre au fil des années sans y insérer les années. La structure de données la plus adaptée en Python est :

1. une liste;
2. un dictionnaire;
3. une pile;
4. une file.

Exercice n° 3 -- 8 pts

On définit un **arbre de Fibonacci** comme un arbre A_n tel que les sous arbres droit et gauche de A_n sont respectivement A_{n-1} et A_{n-2} , en admettant que A_0 est un arbre avec une unique racine et A_1 est composé d'une racine ayant un fils gauche et un fils droit.

Voici une représentation sagittale de l'arbre A_2 .



1. Donner une représentation sagittale de l'arbre A_3 .
2. Quelle est la formule récurrente qui donne la taille d'un arbre en fonction des tailles de ses sous arbres gauche et droit ?
3. Sur votre compte **ReplIt** traiter le projet nommé **Arbres de Fibonacci et taille**.
4. On nomme T_n la taille de l'arbre de Fibonacci A_n .

A l'aide de votre programme Python, donner les valeurs de $T(n)$ pour des valeurs de n comprises entre 0 et 5.

Conjecturer une relation de récurrence entre T_n , T_{n-1} et T_{n-2} .

Exercice n° 4 -- 3 pts

Représentation d'une liste avec un tableau

Une manière simple de représenter une liste est d'utiliser un tableau de taille fixe dont chaque élément est identifié par son indice.

On peut représenter une liste contenant n éléments avec un tableau $L[0..n]$ comme suit :

- La première case du tableau (d'indice 0) contient le nombre d'éléments présents dans la liste.
- Les cases suivantes du tableau (d'indices 1 à n) contiennent les éléments de la liste ou bien sont vides.

(1, 9, 6)



3	1	9	6		
0	1	2	3	4	5

Ecrire en Python (ou en pseudo-code) le code de la fonction `SUPPRIMER(L, i)` qui supprime l'élément à la position i de la liste L .

Exercice n° 5 -- 3 pts

1. Sur quel principe sont basées les **files** ?
2. Donner les opérations de base que l'on peut réaliser sur une **file**.

On donnera le nom des fonctions avec leur(s) paramètre(s) et la description de ce que fait chaque fonction de base.

Exercice n° 6 -- 2 pts

On nomme h la hauteur d'un arbre binaire.

Donner en fonction de h :

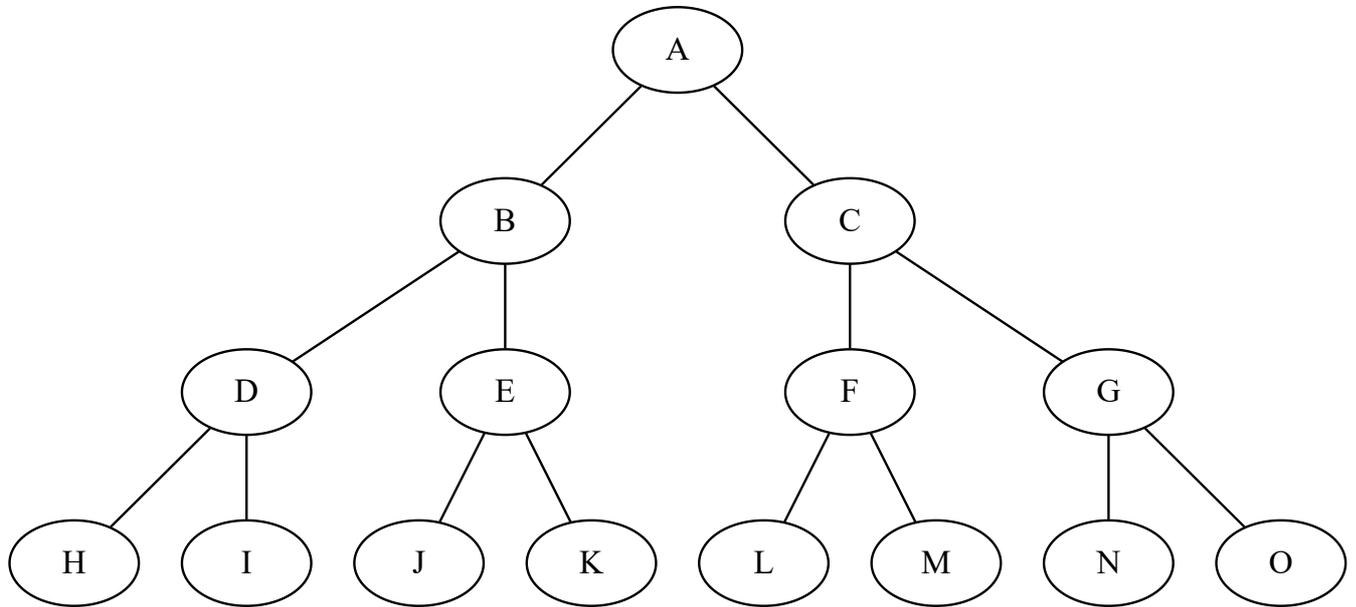
1. La taille minimale d'un arbre binaire.
2. La taille maximale d'un arbre binaire.

Comment nomme-t-on un arbre binaire de taille maximale ?

Exercice n° 7 -- 9 pts

1. Quelles sont les principales caractéristiques d'un arbre ?
2. Donner la définition de la **hauteur** d'un arbre.
3. Donner la définition d'un arbre **dégénéré**.

On considère l'arbre suivant :



4. Donner les caractéristiques de cet arbre.
5. Cet arbre est-il un arbre binaire complet ? Justifier.
6. Calculer (en détaillant) les mesures suivantes de cet arbre :
 - Sa taille
 - Sa hauteur
 - Sa longueur de cheminement : LC
 - Sa longueur de cheminement externe : LCE
 - Sa longueur de cheminement interne : LCI
 - Sa profondeur moyenne : PM
 - Sa profondeur moyenne externe : PME
 - Sa profondeur moyenne interne : PMI

Ranger par ordre croissant les 3 profondeurs calculées.