

CONCOURS BLANC DE JUIN 2024

INFORMATIQUE

AVERTISSEMENT

- On changera de page à chaque début d'exercice.
- La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies.
- Pour améliorer la compréhension, les fonctions en Python doivent être commentées.
- Les résultats non encadrés et non justifiés ne seront pas pris en compte.

Exercice 2 - Etude d'un algorithme

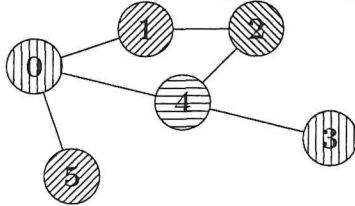
On donne ci-dessous le code (incomplet) d'une fonction de calcul dichotomique d'une racine d'une équation $f(x) = 0$ sur les réels :

```
1 def dicho(f:callable, a:float, b:float, eps=1e-5)->float:
2     """Retourne une approximation de l'unique solution
3     de l'équation f(x)=0 sur l'intervalle ]a,b[.
4     - f est continue et strictement monotone sur ]a,b[
5     - f(a)*f(b)<0
6     - a<b
7     """
8     while b-a>eps:
9         m=
10        if f(m)==0:
11            return m
12        elif f(a)*f(m)<0:
13            b=
14        else:
15            a=
16    return (a+b)/2
```

- 2.1 Compléter les lignes 9, 13 et 15 de la fonction précédente. (Ne réécrire sur la copie que ces lignes 9, 13, 15)
- 2.2 Expliquer le critère d'arrêt de la boucle while de la ligne 8.
Pourquoi n'utilise-t-on pas de critère d'égalité ?
- 2.3 Exprimer le nombre maximum d'itérations de la boucle while en fonction de a, b et eps.
Quelle est alors la complexité de cet algorithme ?
- 2.4 En utilisant l'invariant de boucle "quelle que soit l'itération, la solution cherchée appartient à $]a, b[$ ", démontrer *rigoureusement* que la fonction dicho est correcte.

A Questions préliminaires

3.A.1 On considère le graphe G_2 suivant :



Fournir la matrice d'adjacence, la liste c des couleurs, la liste d des degrés.

- 3.A.2 Ecrire une fonction `degre(matrice:list)->list` qui prend en argument une liste de listes représentant la matrice d'adjacence d'un graphe G et fournit la liste de ses degrés. d
- 3.A.3 Ecrire une fonction `voisins(matrice:list,s:int)->list` qui prend en arguments une liste représentant la matrice d'adjacence d'un graphe G et un sommet s et fournit la liste des voisins de s .
- 3.A.4 En déduire une fonction `PremiereCouleur(matrice:list, s:int, c:list)->int` qui prend en arguments une liste représentant la matrice d'adjacence d'un graphe G , un sommet s et une liste de coloration c et qui renvoie le code de la première couleur (autre que blanc) non utilisée par les voisins de s .
 Dans l'exemple ci-dessus, pour le sommet 4, comme les couleurs utilisées par ses voisins sont 1 et 4, la fonction doit fournir la valeur 2.

B Algorithme naïf

- 3.B.1 Ecrire une fonction `naif(matrice:list)->list` qui prend en argument une liste représentant la matrice d'adjacence d'un graphe G et fournit la liste de coloration en respectant l'algorithme suivant : on examine les sommets dans l'ordre de numérotation et on affecte à chaque sommet la première couleur non utilisée par ses voisins.
- 3.B.2 Si on considère le graphe G_3 suivant, quelle est la coloration fournie par l'algorithme naïf ? Est-ce que le nombre de couleurs est minimal ?

