

1 Autour de l'algorithme de recherche dichotomique

1. Écrire l'algorithme de recherche dichotomique dans une liste triée. Dans la mesure du possible, essayer de l'écrire sans consulter le cours. Tester votre algorithme sur différentes listes triées. On pourra former des listes avec un grand nombre d'éléments choisis au hasard.
2. Modifier légèrement votre algorithme pour qu'il renvoie le nombre de passages dans la boucle *while*. Tracer la courbe donnant ce nombre de passages dans la boucle *while* en fonction de la longueur de la liste choisie qui variera de 0 à 10^4 . On prendra une liste de nombres aléatoires entre 0 et 10^9 et on cherchera un élément qui n'est pas dans la liste (pire des cas). Analyser la courbe obtenue.
3. Dans l'algorithme de recherche dichotomique dans une liste triée, nous avons deux tests à chaque passage dans la boucle *while* : *if* $L[m] == a$ et *elif* $L[m] < a$. On peut accélérer l'algorithme en faisant un unique test d'égalité en sortie de boucle, comme dans la fonction suivante :

```

1  def dico2(L, a):
2      """recherche si l'élément a est présent dans la liste triée dans l'ordre croissant L et renvoie un
   indice si c'est le cas et False sinon"""
3      g = 0
4      d = len(L)
5      if d == 0:
6          return(False)
7      while d > g + 1:
8          m = (g + d) // 2
9          if L[m] > a:
10             d = m
11         else:
12             g = m
13     return(L[g] == a)

```

Écrire et tester ce nouvel algorithme en essayant de comprendre son fonctionnement.

2 Recherche dichotomique d'un zéro d'une fonction

1. Écrire l'algorithme de recherche dichotomique d'un zéro d'une fonction vu dans l'exercice 4 du TD du cours 5.
2. Trouver une valeur approchée de $\sqrt{2}$ et de π à 10^{-14} près.
3. Tracer la fonction $f : x \mapsto \frac{1}{20}(x+4)(x+2)(x+1)(x-1)(x-3)+1$ sur l'intervalle $[-4.5, 3.5]$. En déduire une valeur approchée à 10^{-10} près de ces trois zéros.

3 Stratégie de dichotomie dans un jeu

Il s'agit de simuler un jeu où un nombre entier entre 1 et 1000 est choisi par la machine et le joueur doit deviner ce nombre en un minimum de coups. À chaque étape, le joueur rentre un nombre au clavier et la machine répond en affichant un message : "trop petit", "trop grand" ou "gagné". On affichera également un compteur donnant le nombre de coups joués.

1. Écrire la fonction qui réalise ceci. Il est possible d'interagir avec l'utilisateur en lui demandant un nombre et en le sauvegardant dans la variable *choix* ainsi :

```

1  choix = int(input("Entrer un nombre : "))

```

2. Jouer à votre jeu et tenter de gagner en 10 coups maximum. Quelle stratégie utilisez-vous ?