Partie 1

Compléter le script

On donne une partie du script de recherche dichotomique.

1.1 Compléter ce script au niveau des pointillés

```
def rechDico(L, X):
    """
    @params:
    L: list, contient les valeurs rangees dans l'ordre croissant, de
   type int
    X: int, valeur cherchee
    @return:
    -1 si X n'appartient pas a la liste L
    milieu: indice de X dans la liste L
    """
    gauche = 0
    droite = ... ...  # indice de la borne droite de L
    trouve = False
    while gauche <= droite and ... .... ... ... :
        # On se place au milieu de la liste
        milieu = ... ...
        # il, s'agit d'une division entière
        if L[milieu] == ... :
          trouve = True
        # on arrête la boucle
        elif L[milieu] < X:
          gauche = milieu + 1
        else:
            droite = ... ...
    if not trouve :
        return ...
    return ...
```

1.2 Recherche dichotomique dans une liste L

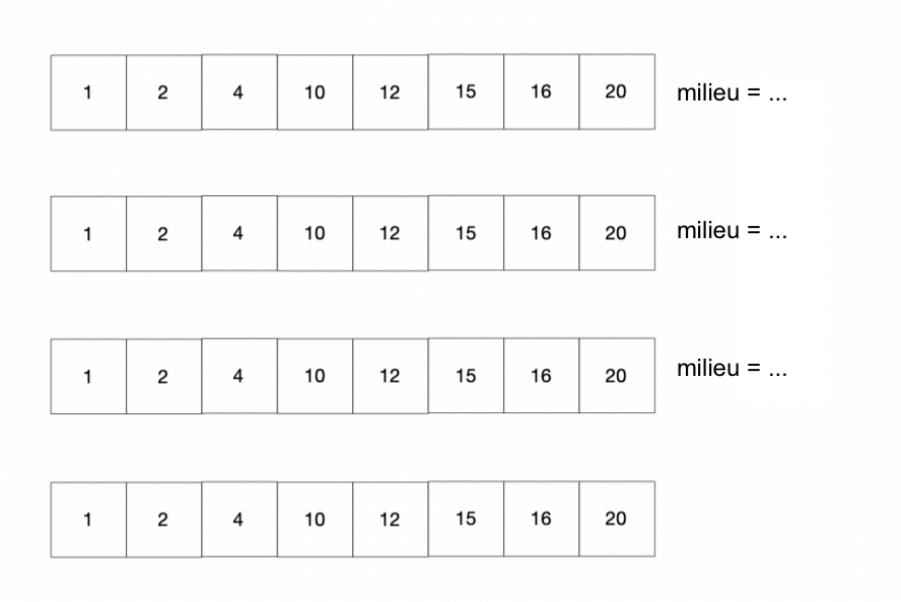

FIGURE 1 – étapes de la recherche dichotomique sur `[1,2,4,10,12,15,16,20]`

1.3 Le document précedent permet le suivi de la recherche dichotomique

1. On execute `rechDico([1,2,4,10,12,15,16,20], 16)`. Colorier pour chaque étape les cases d'indice *gauche* et d'indice *droite* avec 2 couleurs différentes.
2. Compléter la valeur de *milieu* au début de chaque étape.
3. Que retourne la fonction avec l'appel `rechDico([1,2,4,10,12,15,16,20], 16)` ?

1.4 Déterminer la classe de compléxité de cette fonction