

Document 1 - jeu du nombre mystère

Le jeu auquel vous allez jouer est celui d'une recherche dichotomique. Vous allez chercher un élément (une valeur) dans un ensemble trié (l'ensemble des entiers [0 ..100])

```

fonction devine_un_nombre(borne_inf: entier, borne_sup: entier) -> c: entier
  var N: entier[borne_inf .. borne_sup] <- tirage_aleatoire(borne_inf, borne_sup)
  var c: entier <- 0 # compteur du nombre d'essais
  var p: entier[borne_inf .. borne_sup] <- -1 # valeur proposée par l'adversaire
  tant que p != N faire:
    p <- saisie("quel nombre proposes tu?")
    c <- c + 1
    si p > N alors:
      afficher("c'est moins")
    sinon si p < N alors:
      afficher("c'est plus")
    sinon:
      afficher("bravo c'est gagné")
  fintantque
  retourner c

```

Questions: Joue à ce jeu avec ton adversaire.

1. Faire plusieurs parties avec pour bornes: [0 .. 100]. Quelle est la valeur de c dans chaque cas?
2. Quelles sont les puissances m et n de 2 qui encadrent la valeur $2^m < 100 < 2^n$
3. Jouer à ce même jeu, mais avec les bornes: [0 .. 1000]
4. Comparer c avec n et m tels que $2^m < 1000 < 2^n$

Document 2 - Recherche linéaire

```

def recherche_lin(T, x):
  """
  :param T: list of elements
  :param x: element
  :return: int, index of x in the list
           else -1
  """
  i = 0
  while i < len(T) and T[i] != x:
    i = i + 1
  if i >= len(T):
    return -1
  else:
    return i

```

Questions

1. Que retourne l'instruction suivante: `recherche_lin([i for i in range(10)], 5)`? Justifier sommairement.
2. Que retourne l'instruction suivante: `recherche_lin([i for i in range(10)], 10)`? Justifier sommairement.

Document 3 - recherche dichotomique

```
def recherche_dicho(T,x):
    """
    :param T: sorted list of elements
    :param x: element
    :return: int, index of x in the list
             else -1
    """
    i_min = 0
    i_max = len(T) - 1
    while (i_max >= i_min):
        mid = (i_min + i_max)//2
        if T[mid] == x:
            return mid
        elif T[mid] < x:
            i_min = mid +1
        else:
            i_max = mid -1
    return -1
```

Questions

1. Peut-on utiliser cet algorithme pour effectuer une recherche dichotomique dans la liste [0, 2, 5]?
2. Même question pour la liste [10, 2, 5]?
3. Combien d'étapes sont nécessaires pour trouver la valeur 5 dans la liste [0, 2, 5] à partir de cet algorithme? Faire le suivi des variables `i_min`, `i_max` et `mid` au cours de l'avancée du programme.

Document 4 - Tri par insertion

Pour chaque carte de la donne :

Regarder à la fin de la main triée

Mémoriser la clé de cette carte

Tant que la nouvelle carte va avant la carte de la main triée :

Avancer le regard d'une carte vers la gauche dans la main triée

Fin tant que

Insérer la nouvelle carte à gauche de la carte de la main triée qu'on

vient de regarder

Fin pour chaque

Question:

représenter la main du joueur lorsqu'il reçoit les cartes 6, puis 9 puis Valet. Il utilise l'algorithme de tri par insertion pour les classer. Il possède au debut de la partie: 8, 10, Dame, Roi, As

Document 5: Exemple de parcours par dichotomie

On recherche le nom Morin Arthur

1. Comparer les méthodes de recherche séquentielle et dichotomique en suivant pas à pas chacune des étapes par ces 2 algorithmes.
2. Lequel est le plus efficace?

Participants:

Abraham, Max
Abt, Antal
Barlow, Peter
Bartoniak, Géza
Barus, Carl
Bauer, Edmond
Beetz, Johan
Belar, Albin
Blondel, André
Brewster, David
Brillouin, Léon
Dalén, Gustaf
Dolbear, Amos
Duhem, Pierre
Eötvös, Loránd
Fröhlich, Pál
Graetz, Leo
Hall, Edwin
Holten, Carl
Khvolson, Orest
Knudsen, Martin
Küch, Richard
Lamb, Horace
Lebedev, Peter
Lehmann, Otto
Lemoine, Jules
Marsden, Ernest
Morin, Arthur
Perrin, Jean
Poni, Petru
Soret, Charles
Weiss, Pierre
Zeeman, Pieter