Collège de Candolle

ÉPREUVE REGROUPÉE

Discipline: INFORMATIQUE **Date:** 16/12/2022
Cours: 2INDP 01 à 10 **Durée:** 105 minutes

Maître: S Chiou / JC Barros / H Garcia / I Schmidt
Matériel autorisé: aucun (aide-mémoire inclus à la fin de l'épreuve)

Consignes de travail:
- **Ecriture lisible et présentation soignée**
- **Mise en évidence des résultats**

Nom et prénom: **Total: 44 points**

Ex 1: On souhaite trier la suite de nombres suivante: [4, 1, 2, 5, 3] **(3 pt)**

a. Observez les deux premières étapes de l'algorithme de tri ci-dessous et donnez son nom.

Réponse: ……

Étape 1.

```
[4, 1, 2, 5, 3] # On compare 4 et 1 et on les échange.
[1, 4, 2, 5, 3] # On compare 4 et 2 et on les échange.
[1, 2, 4, 5, 3] # On compare 4 et 5 et on ne les échange pas.
[1, 2, 4, 5, 3] # On compare 5 et 3 et on les échange.
[1, 2, 4, 3, 5] # Fin de la première étape.
```

Étape 2.

```
[1, 2, 4, 3, 5] # On compare 1 et 2 et on ne les échange pas.
[1, 2, 4, 3, 5] # On compare 2 et 4 et on ne les échange pas.
[1, 2, 4, 3, 5] # On compare 4 et 3 et on les échange.
[1, 2, 3, 4, 5] # Fin de la deuxième étape.
```

b. Est-ce que l'algorithme de tri est arrivé à son terme? Explicitez votre réponse.

Réponse : ……

Ex 2: Voici toutes les étapes d'un tri de la liste suivante: [1, 3, 2] **(2 pt)**

1. [1, 3, 2]
2. [1, 2, 3]
3. [2, 3, 1]
4. [2, 1, 3]
5. [3, 1, 2]
6. [3, 2, 1]
7. Solution: [1, 2, 3]

Quelle **stratégie algorithmique** a-t-on utilisée pour trouver le résultat ? (justifiez votre réponse)

Réponse : ……

Ex 3: Triez la liste suivante avec l'algorithme de **tri par pivot** (appelé aussi quicksort): **(~5 pts)**

[0, 4, 2, 5, 6, 3, 1]

Indiquez clairement toutes les étapes ci-dessous.
Faites d'abord un brouillon pour que votre réponse soit bien présentée.

Ex 4: On considère le problème de rendu de monnaie optimal **(~3 pts)**
(optimal = avec un minimum de pièces) où l'on doit rendre 10€.
On dispose du système de monnaies imaginaire suivant : 1€, 5€ et 8€.

a. Quel est le rendu de monnaie qu'on obtient avec un algorithme glouton ?
Indiquez clairement le nombre de pièces de chaque sorte.

b. Est-ce la solution optimale ? (oui / non)
Si oui, expliquez votre réponse. Sinon, donnez la solution optimale.

Ex 5: Un Sherpa doit traverser la montagne pour vendre des marchandises **(~6 pts)**
dans le village voisin. Il ne peut pas transporter plus de 20 kg dans son sac à dos
et il dispose de 6 objets de masses différentes et de valeurs différentes.

objets	A	B	C	D	E	F
masses	8	12	3	6	5	2
valeurs	10	8	5	9	7	6

Il se demande quels objets choisir pour transporter la **valeur totale maximale** dans son sac tout en ne dépassant pas 20 kg.

a. Quels objets doit-il mettre dans son sac s'il applique un algorithme glouton dans la résolution de ce problème ?
Complétez le tableau suivant pour répondre.
Attention, il se pourrait qu'il y ait plus de colonnes que nécessaire.

objets	1. …	2. …	3. …	4. …	5. …	totaux
masses						
valeurs						

b. Est-ce la solution optimale ? (oui / non)
Si oui, expliquez votre réponse:
- oui car: .....

Sinon, donnez une meilleure solution dans le tableau ci-dessous:
- non, voici une meilleure solution:

objets	1. …	2. …	3. …	4. …	5. …	totaux
masses						
valeurs						

Ex 6: Le programme suivant doit lister tous les multiples de 5 inférieurs à 100. **(1 pt)**
Corrigez le code pour qu'il le fasse correctement. Pour cela, entourez une erreur et écrivez clairement sa correction.

```
diviseur = 5
for i in range(100):
    if i / diviseur == 0:
        print(i, 'est un multiple de ', diviseur)
```

Ex 7: On souhaite que le programme suivant affiche un décompte de 10 jusqu'à 0. **(2 pt)**
Corrigez le code pour qu'il le fasse correctement. Pour cela, entourez une erreur et écrivez clairement sa correction.

```
1 #Compte à rebours
2 def compte_a_rebours(nombre) :
3     while True :
4         print(nombre)
5         nombre = nombre - 1
6
7 compte_a_rebours(10)
```

Ex 8: Complétez les parties manquantes (en jaune) du programme ci-dessous : **(~6 pts)**

```
# déclarer une liste vide L

L =

# répéter 10 fois

for i

# Demander deux nombre N1 et N2

    N1 =                (“Donnez un nombre décimal : ”)

    N2 =                (“Donnez un autre nombre décimal : ”)

    # identifier le grand nombre et le petit nombre :
    # d'abord, si ce sont les mêmes nombres, l’exécution s’arrête,

    if N1 ... N2 :
        break

    # puis, si les nombres sont différents, on classe les nombres.
    elif N1 < N2 :
        grand_Nombre = N2
        petit_Nombre = N1

    else :

    # Calculer la différence entre ces deux nombres
    difference = grand_Nombre - petit_Nombre

    # ajouter la différence dans la liste L

# fin de répéter

# Calculer la somme des éléments de L.
total = sum(L)

# Calculer la moyenne des éléments de L:

moyenne = total

# fin du programme
```

Ex 9: Voici le pseudo-code d'un algorithme nommé "binary search". **(~10 pts)**
Il reçoit comme entrée une liste ordonnée qu'on appellera **liste** et une valeur qu'on nommera **valeur**. Il a pour but de déterminer si **valeur** se trouve dans **liste**.
Il retourne FAUX si **valeur** n'est pas dans **liste** et VRAI si **valeur** se trouve dans **liste**.

```
DONNEES_ENTREES: liste, valeur

min ← 0
max ← longueur(liste)
trouve ← FAUX
TANT_QUE (min < max) ET (trouve EST_EGAL_A FAUX) {
  milieu ← ENTIER( (min+max)/2 )
  SI liste[milieu] EST_EGAL_A valeur
  ALORS trouve ← VRAI
  SINON {
      SI liste[milieu] < valeur
      ALORS min ← milieu+1
      SINON max ← milieu-1
      }
}

DONNEE_SORTIE: trouve
```

remarque: ENTIER est une fonction qui retourne la valeur entière d'un nombre.

a. Quelle est la stratégie employée dans cet algorithme? **(1 pts)**
réponse:

b. Écrivez la fonction "binary_search" en Python correspondant à cet algorithme; le début (avec les entrées) et la fin (avec la sortie) sont déjà écrits sur la page suivante. Faites un brouillon et recopiez au net car il n'y a pas beaucoup de place. **(~6 pts)**

Fonction Python de recherche dichotomique:

```
def binary_search(liste, valeur):
    # Écris ton code ci-dessous:
    # 1 ligne/cm, indentation = 1cm
    # lignes vides avant return: ok




    return trouve
```

c. Indiquez clairement dans le pseudo-code les modifications permettant de retourner la position (au lieu de **VRAI**) où a été trouvée la **valeur** dans la **liste** ou “-1” (au lieu de **FAUX**) si la **valeur** n’est pas dans la **liste** . **(3 pts)**

Ex 10: Établissez sous forme de pseudo-code ou logigramme (à choix) un algorithme permettant de trouver **la valeur la plus grande** d'une liste désordonnée de nombres. Vous pouvez par exemple commencer par lister les variables nécessaires. Faites un brouillon puis recopiez au net pour soigner la présentation. **(~6 pts)**

FIN DE L'ÉPREUVE (aide-mémoire suit)