Projet 1: Algorithme Tri-fusion

Aurélie & Julie

PROJET

Création d’un algorithme de tri-fusion:

Au lieu d’utiliser la fonction sorted, utiliser cet algorithme pour trier les éléments d’une liste de manière croissante

Etape 1: diviser les éléments d’une liste en plusieurs listes triées de manière croissante

Etape 2: Comparer les premiers éléments de chacune des listes pour identifier les plus petits

Etape 3: Ajouter les plus petits éléments dans une liste finale rangée par ordre croissant

Etape 4: Ajouter un argument reversed avec une condition True / False pour ranger les éléments de la liste par ordre croissant / décroissant

APPROCHE

Etape 1:

Réfléchir aux cas exceptionnels au sein de notre fonction de division de le liste initiale en sous-listes

Etape 2:

Création de la fonction de manière récursive

Etape 3:

Création de la fonction de comparaison des éléments et fusion en une liste rangée par ordre croissant

Etape 4:

Création d'une 3ème fonction pour y ajouter un argument reverse et pouvoir en fonction de cet argument avoir une liste triée en ordre croissant / décroissant

APPROCHE

```
def merge(l_left, l_right):

    final_lst = []

    while len(l_left) > 0 and len(l_right) > 0:
        a = l_left[0]
        b = l_right[0]
        if a <= b:
            final_lst.append(l_left.pop(0))
        else:
            final_lst.append(l_right.pop(0))
    final_lst += l_left + l_right
    return final_lst
```

```
def m_s_asc(lst):
    l_len = len(lst)
    if l_len == 0:
        raise ValueError
    elif l_len == 1:
        return lst
    elif l_len == 2:
        if lst[0] <= lst[1]:
            return lst
        elif lst[0] > lst[1]:
            lst[0], lst[1] = lst[1], lst[0]
            return lst
    else:
        l_left = lst[:l_len//2]
        l_right = lst[l_len//2:]
        l_left = m_s_asc(l_left)
        l_right = m_s_asc(l_right)
        merged = merge(l_left, l_right)
        return merged
```

```
def merge_sort(lst, reversed = False):
    if reversed:
        return m_s_asc(lst)[::-1]
    else:
        return m_s_asc(lst)
```

DIFFICULTÉS

Nous avons rencontré 3 problèmes principaux:

1/ Utiliser la récursivité et pas une boucle « for »

2/ Nous voulions ajouté une condition dans notre fonction def m_s_asc(lst): si un des éléments de la liste lst était autre chose qu'un digit, or après plusieurs essais nous avons remarqué que d'office la fonction renvoyait une erreur

3/ L'ajout de la fonction reverse, nous appelions la fonction à l'infini ce qui nous donnait une erreur de recursion

AMÉLIORER

Nous pouvons améliorer:

- la gestion des cas exceptionnels
- l’optimisation du code
- améliorer nos recherches autonomes (sans JM)