Analyse détaillée du programme Python : Système de gestion interactive des étalages

Introduction

Ce programme Python est un système interactif conçu pour gérer des étalages en fonction des priorités et des totaux associés aux éléments. Il utilise des algorithmes de tri récursif et des structures de données matricielles pour organiser et visualiser les éléments représentant des étalages physiques.

---

Fonctionnalités principales

1. Gestion des priorités et des totaux des éléments :

* - La fonction `donner\_priorité\_A` permet à l'utilisateur de créer une liste contenant les priorités et les totaux des éléments.  
  - Les priorités des éléments sont gérées via l'algorithme de tri par fusion (implémenté dans les fonctions `regler\_le\_groupe\_étape1` et `regler\_le\_groupe\_étape2`).

2. Organisation des éléments dans des matrices :

* - Des matrices sont générées pour représenter les étalages physiques.  
  - La fonction `organisation\_éléments` distribue les éléments dans les matrices en fonction de leurs priorités et de leurs totaux.  
  - Les éléments non attribués sont sauvegardés pour un traitement ultérieur.

3. Mapping des éléments entre les matrices :

* - La fonction `correspendance\_étalage` établit des correspondances entre différents niveaux des matrices pour une gestion plus complexe.

4. Modification des priorités et des totaux :

* - La fonction `modification\_priorité\_élément` permet à l'utilisateur de modifier la priorité d'un élément spécifique.  
  - La fonction `modification\_somme\_des\_categorie` ajuste les totaux des éléments avec des options pour réorganiser ou étendre la liste existante.

5. Visualisation interactive :

* - La fonction `représentation\_visuelle` utilise Plotly pour générer des graphiques interactifs montrant la répartition des éléments dans les matrices.

6. Gestion des éléments restants :

* - La fonction `remplir\_par\_ceux\_qui\_restent` tente de remplir les matrices avec les éléments non attribués, tout en respectant les contraintes de totaux non encore répartis.

---

Algorithmes utilisés

1. Algorithme de tri par fusion (Merge Sort) :

* - Complexité temporelle : O(n log n), où n est le nombre d'éléments dans la liste.  
  - Complexité spatiale : O(n), en raison de la mémoire supplémentaire nécessaire pour les appels récursifs.  
  - Utilisé pour trier les éléments selon leurs priorités, en divisant la liste en sous-listes plus petites jusqu'à une taille de 1, puis en fusionnant les sous-listes triées.

2. Organisation matricielle :

* - Les éléments sont répartis dans les matrices en fonction de leurs totaux.  
  - La complexité temporelle dépend de la taille des matrices et de la distribution des éléments, avec un cas pire de O(n × m), où n est le nombre d'éléments et m la taille des matrices.

3. Réallocation dynamique :

* - Les matrices sont ajustées dynamiquement en fonction des modifications apportées par l'utilisateur, avec une complexité liée aux entrées utilisateur.

---

Structure du programme

- Traitement des entrées : L'utilisateur saisit des données via la ligne de commande, y compris le nom des éléments, leurs priorités et leurs totaux.

- Organisation des données : Les éléments sont répartis dans différentes matrices en fonction de leurs priorités et de leurs totaux, à l'aide de tri récursif et de matrices dynamiques.

- Opérations interactives : Plusieurs options sont fournies pour modifier les données, redistribuer les éléments ou générer une visualisation.

- Résultat : Des graphiques interactifs sont générés via Plotly pour montrer la répartition des étalages.

---

Conclusion

Ce programme met en œuvre un système de gestion des étalages complet, combinant tri récursif et techniques de répartition matricielle. Il est adapté à la gestion des tâches de distribution d'éléments basées sur des priorités.