

Lab Complémentaire : Algorithmes de Tri

1. Lecture de l'image

Ecrire une fonction qui charge et retourne une image (en noir et blanc) depuis un chemin donné.

`def charger(path) : return image`

Visitez la fonction `imread` de `scipy`.

2. D'une image, un tableau

La fonction `imread` de `scipy` retourne l'image comme étant un tableau de lignes. Ecrire une fonction qui, à partir de ce tableau multidimensionnel, retourne en un tableau unidimensionnel obtenu en concaténant les lignes les une après les autres dans l'ordre.

3. D'un tableau, une image

Ecrivez une fonction qui prend en paramètre un tableau et une dimension entière et retourne un tableau multidimensionnel (similaire à celui utilisé pour les images par le module `scipy`) dont la taille des lignes est celle spécifiée par la dimension entière.

4. Sauvegarde

Ecrivez une fonction qui prend en paramètre un tableau multidimensionnel et un chemin, et sauvegarde le tableau comme une image à l'adresse spécifiée.

5. Tri avec des screenshots

Réécrivez la fonction qui implémente la méthode de tri que vous aurez choisi en y incluant un checkpoint qui sauvegarde le tableau trié en tant qu' image après chaque N itérations de tri. N fixé.

6. Lancement

Agencez les fonction implémentée afin de réaliser l'algorithme suivant :

Image = Charger(chemin)

Tableau = Transformer_Image_Tableau(Image)

Checkpoint = N

Tant que Tri _en_cours :

 Trier(Tableau)

 Checkpoint -= 1

 Si Checkpoint == 0 :

 Sauvegarder_image(

 Transformer_Image_Tableau (Tableau) ,

 chemin_de_image

)

NB :

- Choisir N très grand (de l'ordre de 50000 voire plus)
- Faire attention a ne pas changer le chemin de sauvegarde à chaque itération