

Dans cette question, on souhaite écrire une fonction Python d'en-tête `def tri(T)`: permettant de trier dans l'ordre croissant les valeurs d'un tableau unidimensionnel `T` (de type `numpy.array`) sans utiliser de fonction de tri prédéfinie dans Python.

Pour cela, on utilise un algorithme de tri appelé *tri par insertion* et défini de la façon suivante :

— On agit sur les coefficients du tableau `T`, que l'on suppose numérotés de 0 à $s - 1$.

— Pour chaque k compris entre 1 et $s - 1$, avant l'étape k les coefficients numérotés 0 à $k - 1$ de `T` ont déjà été classés dans l'ordre croissant, les coefficients suivants étant inchangés. L'étape k consiste alors à insérer la valeur x du coefficient numéroté k de `T` en bonne position parmi les $k + 1$ premiers coefficients de `T` afin qu'à l'issue de l'étape k les coefficients numérotés 0 à k de `T` soient classés dans l'ordre croissant, les coefficients suivants étant inchangés.

Par exemple en partant du tableau `T` dont le contenu est `[38 28 35 24 31 15]`, les étapes de l'algorithme de tri par insertion donnent successivement :

`[38 28 35 24 31 15]`

`[28 38 35 24 31 15]`

`[28 35 38 24 31 15]`

`[24 28 35 38 31 15]`

`[24 28 31 35 38 15]`

`[15 24 28 31 35 38]`

Le tri doit être « en place », c'est à dire que le tableau se retrouve trié en fin de fonction au lieu de retourner un autre tableau, version triée du premier.

Écrire la fonction `tri` demandée en mettant en œuvre l'algorithme de tri par insertion décrit ci-dessus.