



W1 - Algorithmie

W-ALG-502

Bootstrap pushswap

Découvertes des algorithmes de tri



BUBBLE SORT

Le premier algorithme à coder est le **bubble sort**, un algorithme de tri très commun. Il consiste à échanger l'emplacement des éléments dans un tableau.

Ecrivez une fonction qui permet d'échanger deux éléments dans un tableau à des index donnés, voici le prototype de la fonction:

```
function swap_elem(Array &$array, int $index1, int $index2);
```

Voici un exemple d'usage de cette fonction:

```
$array = [1, 2, 3, 4];
swap_elem($array, 1, 2);

// $array vaut [1,3,2,4]

$array2 = [3, 5, 3, 8, 9];
swap_elem($array2, 2, 4);

// $array2 vaut [3, 5, 9, 8, 3]
```

Tri: Écrivez une fonction qui utilise la fonction précédent pour faire un bubble sort d'un array

```
function bubble_sort_array(Array &$array);
```

Voici un exemple d'usage de cette fonction:

```
$array = [1, 3, 4, 2];
bubble_sort_array($array);

// $array vaut [1,2,3,4]

$array2 = [3, 5, 3, 8, 1];
bubble_sort_array($array2);

// $array2 vaut [1, 3, 3, 5, 8]
```

Pensez à tester cette fonction.



INSTRUCTIONS

Nous allons commencer à découvrir les instructions du projet.

- **swap**: Echange le 2 premiers éléments de la liste (**sa** et **sb** dans le projet)
- **rotate_left**: Prends le premier élément de la liste et le met à la fin de la liste (**ra** et **rb** dans le projet)
- **rotate_right**: Prends le dernier élément de la liste et le met au début de la liste (**rra** et **rrb**)

Dans un premier temps, utilisez uniquement les instructions ci-dessus pour trier manuellement la liste suivante: 15, 8, 42, 4, 16, 23



Cette première étape est uniquement là pour vous aider à comprendre les instructions

Maintenant écrivez une fonction qui reçoit un tableau en paramètre. Cette fonction triera le tableau uniquement à l'aide des instructions précédentes, elle affichera également les instructions utilisées.

Prototype de la fonction:

```
function my_amazing_sorter(Array &$array);
```

Exemple d'utilisation:

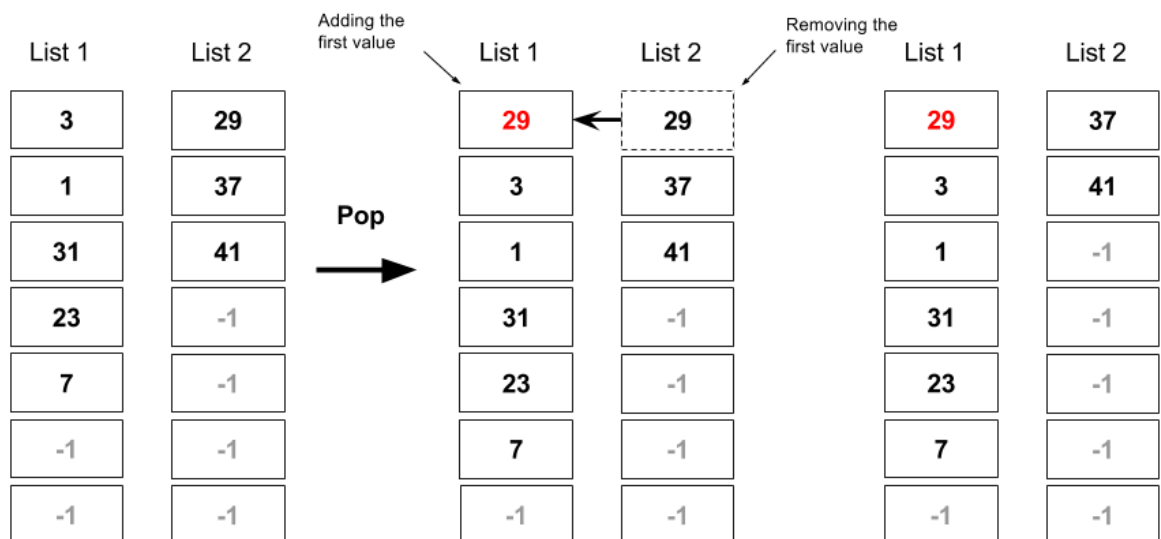
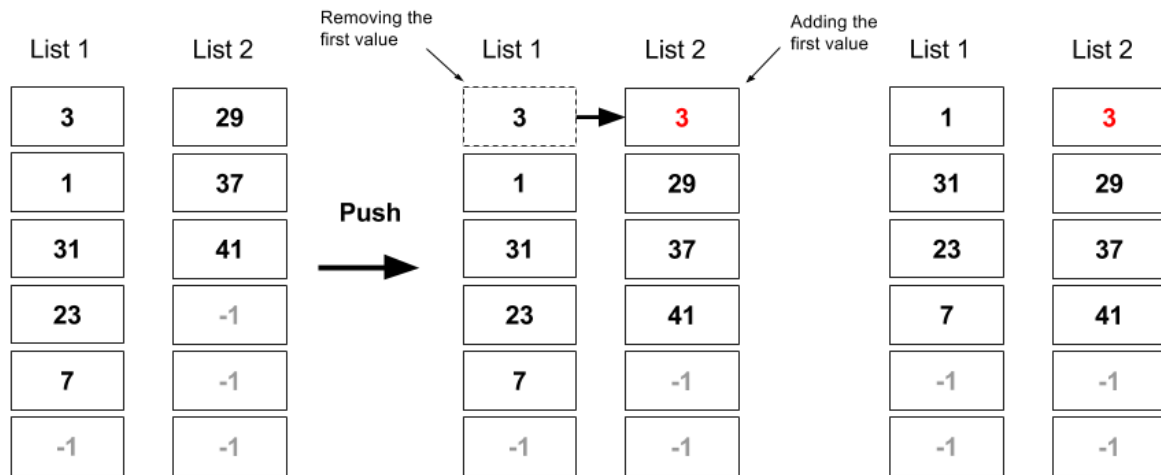
```
$array = [15, 8, 42, 4, 16, 23];  
my_amazing_sorter($array);
```

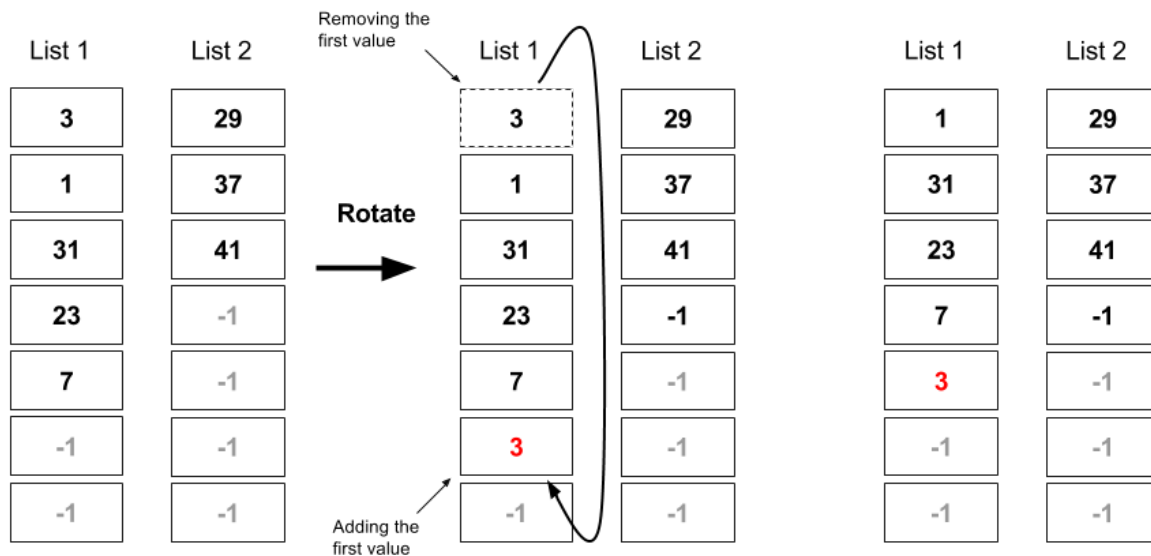
AVEC DEUX LISTES

Maintenant ajoutons les tris avec deux listes

Le challenge var être de créer un algorithme de tri basé uniquement sur les instructions suivantes:

- **push**: Supprime la première valeur de LIST1 et le met au début de LIST2
- **pop**: Supprime la première valeur de LIST2 et le met au début de LIST1
- **rotate**: Supprime la première valeur de LIST1 et le met à la fin de LIST2





Le projet vous donnera plus d'operations que ces 3 opérations

Ecrivez une fonction qui trie une liste uniquement basé sur les opérations ci-dessus, voilà son prototype:

```
function sort_numbers(Array &$array1, Array &$array2);
```

Initialement, le premier tableau est rempli avec des nombres et le 2eme est vide. Le résultat attendu est le premier tableau ordonné.

RESOURCES UTILES

- <https://visualgo.net/en/sorting>