



Algorithmique des tableaux (C5-160211-INFO)

Licence 1 - Année 2019/2020

TP 4 - Ensemble

C. Saint-Jean, L. Mascarilla, E. Zahzah



Le langage *Python* met à disposition une collection appelée *set* pour représenter un ensemble de valeurs. Celle-ci est basée sur une [technique de hachage](#) qui sort du cadre de ce cours de 1^{ère} année. On se propose toutefois de refaire les mêmes fonctionnalités à l'aide d'un *tableau trié*.

Toutes les fonctions que vous allez écrire ⁵ prendront comme premier argument un tableau trié sans doublon.

Certaines fonctionnalités à réaliser correspondent clairement à des algorithmes déjà faits en TP ou vus en cours. Afin de garantir à tous une implémentation sans faute de la dichotomie, nous vous d'utiliser le module [bisect](#) vu en cours.

Liste des fonctions à réaliser

Fonctionnalité	Nom de fonction	Équivalent dans <i>set</i>	Commentaire
Création ⁶	<i>set_create</i> (<i>L</i>)	<i>set</i>	Tri avec suppression des doublons
Taille	<i>set_len</i> (<i>E</i>) → <int>	<i>len</i>	Nombre d'éléments
Appartenance	<i>set_in</i> (<i>E</i> , <i>x</i>) → <bool>	<i>in</i>	Vrai ssi $x \in E$
Non-Appartenance	<i>set_not_in</i> (<i>E</i> , <i>x</i>) → <bool>	<i>not in</i>	Vrai ssi $x \notin E$
Ajout	<i>set_add</i> (<i>E</i> , <i>x</i>)	<i>add</i>	Ajoute un élément. Ne fait rien s'il y est déjà présent.
Suppression	<i>set_rm</i> (<i>E</i> , <i>x</i>)	<i>remove</i>	Supprime un élément. Ne fait rien s'il n'est pas présent.
Disjoint	<i>set_isdisjoint</i> (<i>E1</i> , <i>E2</i>) → <bool>	<i>isdisjoint</i>	Vrai ssi $E1 \cap E2 = \emptyset$
Sous-ensemble	<i>set_issub</i> (<i>E1</i> , <i>E2</i>) → <bool>	<i>issubset</i>	Vrai ssi $E1 \subseteq E2$
Super-ensemble	<i>set_issup</i> (<i>E1</i> , <i>E2</i>) → <bool>	<i>issuperset</i>	Vrai ssi $E1 \supseteq E2$
Union	<i>set_union</i> (<i>E1</i> , <i>E2</i>) → <set>	<i>union</i>	$E = E1 \cup E2$
Intersection	<i>set_inter</i> (<i>E1</i> , <i>E2</i>) → <set>	<i>intersection</i>	$E = E1 \cap E2$
Différence	<i>set_diff</i> (<i>E1</i> , <i>E2</i>) → <set>	<i>difference</i>	$E = E1 \setminus E2$
Différence Symétrique	<i>set_xor</i> (<i>E1</i> , <i>E2</i>) → <set>	<i>symmetric_difference</i>	$E = (E1 \setminus E2) \cup (E2 \setminus E1)$