

TP (solo, noté) La classe `DynArray` (2)

Algorithmique – Programmation FIP (ING39)

V. Aponte, P. Courtieu

Ce qui vous est demandé

Vous devez ré-implanter la classe des tableaux dynamique dans une nouvelle classe : `tpdynarray.DynArray2` (utilisez *exactement* ce nom de classe) dans le même projet `git` que dans le TP précédent. Vous utiliserez le même jeu de test que pour la première implantation : un jeu de tests est indépendant de l'implantation choisie. Vous pouvez bien sûr ajouter des tests en plus. N'oubliez pas de 1) tester, 2) commiter 3) pousser régulièrement. N'attendez pas la dernière minute. AUCUN COMMIT AU DELÀ de l'heure limite du TP ne sera accepté, même de bonne foi.

Contraintes d'implantation

Il existe plusieurs solutions d'implantation. La solution naïve et *inefficace* consiste à stocker *la totalité du préfixe non vide* du tableau infini. C'est la solution qui vous a été imposée au TP précédent. L'inconvénient de cette solution est qu'elle stocke non seulement les cases non vides, mais aussi celles contenant 0. Il peut y en avoir beaucoup lorsque deux indices non vides sont très éloignés. Par exemple, supposons qu'on ait uniquement rempli la case d'indice 100000 avec la valeur 5. Après modification, la liste de notre implantation comporte 100001 cases, où les 100000 premières contiennent 0, et seule la dernière contient 5.

indices :	0	1	2	...	99000	100000
tab :	0	0	0	...	0	5

Il s'agit cette fois de ne pas stocker les cases vides. Pour cela il vous faut stocker non seulement les valeurs non vides *mais également leur positions*. Vous devrez utiliser une (ou plusieurs) variables d'instance de type `ArrayList` pour représenter le tableau infini.

Il ne vous est pas demandé de rendre vos jeux de test (mais ce n'est pas grave si ils sont dans votre archive). La note sera fonction de l'implantation effective d'une solution non naïve et du nombre de tests réussis parmi ceux que nous avons élaboré et la qualité de votre implantation.