

TP 5 - Questions

Vincent Therrien 1956352

Derek Ojeda Centeno 1843863

1. Quel est l'avantage d'utiliser une linked-list au lieu d'un vecteur si on veut supprimer ou rajouter un élément ? Quelle structure de données est plus avantageuse si on veut accéder à des éléments à des positions aléatoires ?

Il y a un gain de vitesse d'exécution lorsqu'on insère un élément à une position autre que la fin du conteneur car la liste est organisée différemment, soit par liens contrairement au vecteur où les éléments doivent se suivre dans la mémoire. Alors, insérer dans un liste ne nécessite que de modifier 2 ou 4 pointeurs, ce qui est plus rapide qu'effectuer une copie pour chaque élément à déplacer lors de l'insertion/la suppression dans un vecteur. Dans le cas de l'accès aléatoire, le vecteur est plus avantageux, car celui-ci surcharge l'opérateur `[]` et permet d'accéder une valeur à l'aide d'un indice qui représente sa position. Cependant, pour la liste, l'accès à une valeur autre que celle de début ou de fin nécessite de parcourir la liste à partir du début jusqu'à trouver l'élément en question.

2. Pourquoi est-ce que l'implémentation des classes génériques est dans `.h` et non pas séparée en `.h` et `.cpp` comme les classes normales ?

Le compilateur doit être en mesure de vérifier si les méthodes appelées sur le types génériques sont effectivement déclarées pour les types spécifiés dans l'utilisation des classes génériques. Par exemple, dans l'implémentation de la classe générique, s'il y a utilisation d'une surcharge d'opérateur, la présence de l'implémentation dans le `.h` permet au compilateur de s'assurer que la surcharge est bien définie pour les types utilisés par la suite.