

TP2 : Opérateurs

Question 1 :

Comme dans le TP1, les tests sont tous regroupés dans le même fichier main, qui se trouve dans src

Question 2 :

Concernant le premier opérateur : `Dvector operator+ (Dvector a, Dvector b)`, nous faisons juste une somme de deux vecteurs qui sont libres dans leur état

Pour le second : `Dvector operator+ (const Dvector &a, const Dvector &b)`, les deux arguments sont des const, on ne peut donc pas les modifier, ils sont protégés.

Regardons maintenant le nombre d'objets créés : Pour le premier cas, nous créons et supprimons 5 vecteurs, car nous créons et détruisons des espaces mémoire liés aux vecteurs.

En ce qui concerne le second cas, nous créons et détruisons seulement 4 vecteurs car nous travaillons plus souvent sur les pointeurs que sur les vecteurs en eux même

Question 3 :

Nous avons essayé de créer et d'utiliser const quand cela était nécessaire, et nous avons commenté les endroits où il pouvait y avoir une ambiguïté dans notre code.

Question 4 :

Nous pouvons ici factoriser deux parties dans le code

La première est celle concernant les opérations liant un vecteur et un double, au lieu de créer les deux opérations pour par exemple faire `v = v + 2`, nous avons créé l'opérateur `+=` qui permet de n'écrire que `v += 2` et donc de factoriser un bout de code.

La seconde factorisation possible est celle concernant les opérateurs d'affectation.

Question 5 :

Analysons les temps de calcul de chaque méthode :

Sans le copy, nous avons les temps suivants :

- real 0m0.080s
- user 0m0.068s
- sys 0m0.010s

Avec le copy, nous avons les temps suivants :

- real 0m0.013s
- user 0m0.11s
- sys 0m0.008s

Avec le memcpy, le programme s'exécute plus rapidement de manière significative, néanmoins, nous conservons la version classique car elle est plus simple d'utilisation et nous avons quelques erreurs avec le memcpy que nous n'avons pas résolues

Question 6 :

Comme pour le TP1, nous avons testé l'ensemble des fonctions et des allocations dans le main, la mémoire allouée et libérée était donc la même que lors de la réponse au TP1, c'est à dire que toute la mémoire allouée était libérée.