

Cours 5 - TD

Résumé

L'objectif de ce TD est d'implémenter une structure de données pour gérer des ensembles ainsi que les opérations ensemblistes sur celle-ci. Pour cela, nous allons implémenter une liste chaînée qui va nous servir pour stocker les valeurs de l'ensemble.

Implémentation d'une liste

Une liste contient des éléments faits d'une valeur et d'un pointeur sur l'élément suivant. On souhaite disposer des opérations suivantes : ajout d'un élément (du type de la valeur d'un élément) et recherche d'un élément (renvoyer un booléen).

Définir une structure pour la liste (comme dans le TD du cours 4) en utilisant des entiers. Implémentez les 2 opérations en tant que méthodes (c'est à dire dans la structure). Pensez que le marqueur de fin de liste est le pointeur 0 et que l'ajout nécessite une allocation dynamique. Nous considérerons que le premier élément de la liste est une tête fictive, c'est à dire que sa valeur ne nous intéresse pas. Ainsi une liste vide ne possédera qu'un élément. Vous prendrez garde à vous assurer que le dernier élément d'une liste a toujours son pointeur sur l'élément suivant à 0 ...

Définition d'un ensemble

Définissez une structure possédant une liste pour stocker les valeurs d'un ensemble. Vous définirez les méthodes suivantes : ajout d'une valeur (attention à garantir l'unicité!) et recherche d'une valeur (renvoyer un booléen). Vous surchargerez les opérateurs \ll et \gg pour écrire/lire les valeurs vers/dans un flux quelconque (cout, cin, fichier, ...).

Définissez des fonctions pour calculer l'union, l'intersection et la différence entre 2 ensembles. Les 2 ensembles constitueront les paramètres de chaque fonction qui renverra un nouvel ensemble contenant le résultat.

Pour finir, tout passer en templates.