TD9-ω

1 - Listes chaînées (encore …)

Une façon efficace d'implémenter une file (FIFO) consiste à utiliser deux listes simplement chaînées. La première liste sert au retrait d'éléments. La seconde sert à l'ajout. Lorsque la première est vide, on la remplace par la seconde liste inversée, et celle ci est remplacée par une liste vide. Ainsi, l'ajout (en tête dans la seconde liste) et le retrait (en tête dans la première liste) dans la file se font en temps constant, à par le moment où il est nécessaire d'inverser la deuxième liste. Il existe une théorie pour analyser la complexité d'une telle implémentation qui s'appelle analyse amortie qui est relativement complexe, mais on peut intuitivement mesurer son gain : au lieu de "payer" systématiquement O(n) opérations pour ajouter un élément, on ne "paye" O(n) opérations qu'au moment de l'inversion.

Dans la suite, on supposera qu'on dispose d'une structure de liste chaînée, avec les fonctions de base qui permettent d'ajouter en tête, de retirer en tête, d'inverser sur place la liste, et de savoir si la liste est vide.

a) définir la structure de file
b) écrire une fonction qui permet d'ajouter un élément à la file
c) écrire une fonction qui permet de retirer un élément à la file

2 - Calcul du coefficient binomiale

Ecrire une fonction qui prend en paramètre deux entiers n et p et qui renvoie le coefficient binomial $\binom{n}{p}$.

Rappel : le coefficient binomial peut se calculer comme ceci :

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}$$

Mais il est plus efficace de le calculer récursivement (formule de Pascal):

$$\binom{n+1}{p+1} = \binom{n}{p+1} + \binom{n}{p}$$