
OPTION INFORMATIQUE

TP n°2 : listes, tableaux

Exercice 1. *Concaténation simple, renversement.*

On rappelle que le `cons` est l'opération consistant à former une nouvelle liste en ajoutant un élément en tête d'une liste.

1. En cours, on a réécrit la fonction `List.concat` et c'était dur.
Faisons plus simple : réécrivons la fonction `List.append`, plus simplement appelée `(@)`.
Combien de `cons` sont nécessaires pour concaténer deux listes `l1` et `l2`?
2. Réécrivons maintenant la fonction `List.rev` qui permet de renverser une liste.
3. Combien de `cons` sont nécessaires pour que votre fonction renverse une liste `l` de longueur `n`?
Si c'est strictement plus que `n`, c'est trop, recommencez!

Exercice 2. *Binomiaux.*

On rappelle que les coefficients binomiaux $\binom{n}{k}$ vérifient les propriétés suivantes :

- $\forall n \in \mathbb{N}, \binom{n}{0} = 1$;
- $\forall k > 0, \binom{0}{k} = 0$;
- $\forall (n, k) \in \mathbb{N}^2, \binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$.

1. Écrire une fonction récursive permettant de calculer les binomiaux à l'aide de ces relations.
2. Écrire une fonction **itérative** réalisant la même chose par mémoïsation.
On utilisera une matrice (= tableau de tableaux, qu'on peut initialiser avec `Array.make_matrix`) d'entiers permettant de stocker successivement les valeurs des $\binom{i}{j}$ avec $0 \leq i \leq n$ et $0 \leq j \leq k$.
3. Laquelle des deux versions est la plus rapide? Oui, on s'en doute normalement, mais vérifiez tout de même à l'aide de la fonction `Sys.time` qui permet de mesurer les temps d'exécution.