TP 9 informatique

BCPST 1 2019-2020

Les listes

Exercice 1. Écrire une fonction `permut_aleatoire(n)` qui prend en argument un entier $n \geq 0$ et qui renvoit une n-liste sans répétition de $\{0, 1, \ldots, n-1\}$ aléatoire. On rappelle qu'il est possible de tirer un nombre aléatoire. Par exemple, si on écrit :

```
import random
indice = random.randint(0,10)
```

dans `indice` contient une valeur tirée au hasard dans $\{0, 1, \ldots, 10\}$.

Exercice 2. Écrire une fonction `est_sans_repet(L)` qui prend en argument une liste `L` et qui renvoit `True` si `L` est sans répétition et `False` sinon.

Exercice 3. Soient $n \geq 1$, L une n-liste sans répétition de $\{0, 1, 2 \ldots, n-1\}$. Soit $i \in \{0, 1, \ldots, n-1\}$. On dit que i est point fixe de L si $L[i] = i$. Écrire une fonction `nb_points_fixes(L)` qui prend en argument une liste sans répétition de $\{0, 1, 2, \ldots, len(L) - 1\}$ et qui renvoit le nombre de points fixes de L.

Exercice 4. Pour tout $n \in \mathbb{N}$, on note $P(n) = \{\text{partitions de } \{1, 2, \ldots, n\}\}$ et p_n le cardinal de $P(n)$. Par convention, $p_0 = 1$.

1. Écrire toutes les partitions de $\{1, 2, 3\}$.
2. (math dur, facultatif) Montrer que pour tout $n \in \mathbb{N}^\star, p_{n+1} = \sum_{k=0}^{n} \binom{n}{k} p_{n-k}$.
3. En déduire une fonction `nb_partitions(n)` qui prend en argument un entier n et qui renvoit la liste contenant $[p_0, p_1, \ldots, p_n]$.

Exercice 5. (Si vous avez le temps) Étant une n-liste sans répétition L de $\{0, 1, 2, \ldots, n-1\}$, il est possible de construire la liste inverse M de taille n définie par

$$\forall (i, j) \in \{0, 1, 2, \ldots, n-1\}^2, L[i] = j \iff i = M[j].$$

Écrire une fonction `calcul_inverse(L)` qui prend en argument une liste sans répétition L et qui renvoit sa liste inverse.

Exercice 6. (Si vous avez le temps)

1. Étant donnée liste de n entiers L, une inversion de L est un couple $(i, j) \in \{0, 1, \ldots, n-1\}^2$ vérifiant $i < j$ et $L[i] > L[j]$. Écrire une fonction `listes_inv(L)` prenant en argument une liste L et qui renvoit la liste des inversions de L.
2. (math dur) Déterminer le nombre de listes de taille n sans répétition de $\{0, 1, \ldots, n-1\}$ ayant exactement 1 inversion.
3. (math dur) Montrer qu'une liste de taille n sans répétition est triée dans l'ordre croissant si et seulement si elle n'a pas d'inversion.