

## 1 Tris

**Question 1.** Écrire les fonctions `tri_fusion` et `tri_rapide` de type `int list -> int list`.

## 2 Arbres binaires

---

```
type 'a arbre = F of 'a | N of 'a * 'a arbre * 'a arbre;;
```

---

**Question 2.** Écrire une fonction `abr` qui teste si un arbre binaire est un arbre binaire de recherche.

**Question 3.** Écrire une fonction `max_sum` qui prend en argument un arbre binaire étiqueté par des entiers et renvoie la valeur maximale de la somme des éléments d'une de ses branches.

**Question 4.** Écrire une fonction `préfixe` qui prend en argument un arbre binaire étiqueté par des caractères et renvoie son ordre préfixe. Faire de même pour `infixe`.

## 3 Nombres

**Question 5.** Écrire des fonctions `pgcd` et `ppcm` qui déterminent le PGCD  $a \wedge b$  et le PPCM  $a \vee b$  de deux entiers  $a$  et  $b$ .

**Question 6.** Écrire des fonctions `encode` et `decode` qui convertissent un nombre en base quelconque depuis et vers la base décimale.

## 4 ★

**Question 7.** Écrire une fonction `bézout` qui prend en argument deux entiers  $a$  et  $b$  et retourne deux entiers  $u, v \in \mathbb{Z}$  tels que  $au + bv = a \wedge b$ .

**Question 8.** Écrire une fonction qui à partir des ordres préfixe et infixe d'un arbre binaire, donnés sous la forme de chaînes, reconstruit l'arbre.

**Question 9.** Écrire une fonction `embrouillage` qui prend en argument un entier  $p$  et retourne un vecteur `miroir` à  $n = 2^p$  éléments tel que `miroir.(i)` est le miroir de  $i$  sur  $p$  bits.

Par exemple, si  $p = 5$ , comme  $11 = 01011_2$ , `miroir.(11) = 11010_2 = 26`.

Votre fonction devra être en  $O(n \log n)$ , ou même en  $O(n)$ .

## 5 Oups !

Yorel Reivax tente de coder la fonction `max3`, qui retourne le maximum de 3 éléments.

---

```
let max3 a b c = max max a b c;;
```

---

Cam1 retourne `max3 : ('a -> 'a -> 'a) -> 'a -> 'a -> 'a = <fun>`, pourquoi ?  
Déterminer un appel de `max3` qui ne retourne pas d'erreur.