TD3 : PF.

Exo 1.

1. monMinimum :: (Ord a) => [a] -> Maybe a
2. monMinimum [] = Nothing
   monMinimum [a] = Just a
   monMinimum (x1 : x2 : xs)
   | x1 < x2 = monMinimum (x1 : xs)
   | otherwise = monMinimum (x2 : xs)

Exo 2.

Q2.1
```
data ABR a
  = Noeud a (ABR a) (ABR a)
  | Vide
```

Q2.2
```
inserer :: (Ord a) => a -> ABR a -> ABR a
inserer v Vide = Noeud a Vide Vide
inserer v (Noeud v' g d)
    | v < v' = inserer g v
    | otherwise = inserer d v
```

Q2.3
```
abrToList :: (Ord a) => ABR a -> [a]
abrToList Vide = []
abrToList (Noeud v g d) = (abrToList g) ++ [v] ++ (AbrToList d)

estOrdonne :: Ord a => [a] -> Bool
estOrdonne [] = TRue
estOrdonne [_] = TRUe
estOrdonne (x1:x2:xs) = x1 < x2 && estOrdonne xs

estABR :: Ord a => ABR a -> Bool
estABR a = estOrdonne abrToList a
```

Exercice 4

Q1.
```
hauteur :: Ord a => ABR a -> Int
hauteur Vide = -1
hauteur (Noeud _ g d) = 1 + max $ hauteur g $ hauteur d
```