

## Lista 9

1. Rozważ poniższy typ dla drzew binarnych:  
data Tree a = Leaf a | Node (Tree a) (Tree a)  
deriving (Eq,Ord,Show,Read)

Drzewo jest zrównoważone, jeśli liczba liści w lewym i prawym poddrzewie każdego węzła różni się co najwyżej o 1 (to jest jedna z możliwych definicji). Zdefiniuj funkcję `isBalanced :: Tree a -> Bool`, która sprawdza zrównoważenie drzewa.

2. Zdefiniuj funkcję `balance :: [a] -> Tree a`, konwertującą niepustą listę w drzewo zrównoważone.

3. W regularnym drzewie binarnym każdy z węzłów jest bądź liściem, bądź ma stopień dwa (patrz Cormen i in. §5.5.3). Rozważmy następujący typ dla drzew binarnych:

```
data BTree a = BLeaf | BNode (BTree a) a (BTree a)
deriving (Eq,Ord,Show,Read)
```

Długość ścieżki wewnętrznej  $i$  regularnego drzewa binarnego jest sumą, po wszystkich węzłach wewnętrznych drzewa, głębokości każdego węzła. Długość ścieżki zewnętrznej  $e$  jest sumą, po wszystkich liściach drzewa, głębokości każdego liścia. Głębokość węzła definiujemy jako liczbę krawędzi od korzenia do tego węzła.

Napisz dwie funkcje, obliczające dla zadanego regularnego drzewa binarnego:

- a) długość ścieżki wewnętrznej,
- b) długość ścieżki zewnętrznej.

4. Zdefiniowany jest następujący typ dla drzew wielokierunkowych:

```
data MtreeL a = MTL a [MtreeL a]
deriving (Eq,Ord,Show,Read)
```

Napisz dwie funkcje, obchodzące drzewa wielokierunkowe:

- a) w głąb prefiksowo (preorder),
- b) wszerz.