

Warunki

1. Zadania można wysyłać, jeśli ktoś ma najwyżej 2 plusy (licząc od 18 III).
2. W implementacjach można korzystać tylko z elementarnych konstrukcji Python'a (funkcje, instrukcje warunkowe, pętle, `range`, klasy użyte do definiowania struktur danych, wbudowana funkcja sortująca, itp.). **Nie wolno korzystać ze słowników i zbiorów.**
3. Rozwiązania muszą być efektywne obliczeniowo (także w zadaniach, w których nie podajemy wprost ograniczenia na złożoność obliczeniową). Zadania o zbyt wysokiej złożoności będą oceniane na brak plusa.
4. **Wolno omawiać zadania (w tym pomysły na implementację), ale wyłącznie na forum w systemie UPEL. Nie wolno wymieniać kodu realizującego fragmenty algorytmu (ale wolno odpowiadać na pytania postaci "jak zrealizować tablicę dwuwymiarową" itp.).**

Zadanie 1 (BST)

Dane są klasy opisujące słownik zrealizowany jako drzewo BST:

```
# wezeł drzewa BST
class BSTNode:
    def __init__(self, key, value):
        self.left = None
        self.right = None
        # pola parent nie ma -- nie jest w tym zadaniu potrzebne
        self.key = key
        self.value = value

class BSTDict:
    def __init__( self ):
        self.tree = None      # tu powinien być korzeń drzewa
                              # gdy słownik nie jest pusty

    def insert( self, key, value )
        # wstaw wartość value pod klucz key (jeśli klucz key
        # już istnieje, to podmień przechowywaną wartość
        # value

    def remove( self, key ):
        # usuń z drzewa węzeł z kluczem key
```

Proszę zaimplementować funkcje `insert` oraz `remove` z klasy `BSTDict` (wygodne może być także zaimplementowanie funkcji wyszukiującej klucza w drzewie lub wypisującej drzewo, ale nie jest wymagane). Państwa kod będzie wykonywany.

Zadanie 2 (Poprawność drzew czerwono-czarnych)

Dana jest klasa opisująca węzeł drzewa czerwono-czarnego:

```
BLACK = 0
RED   = 1

class RBNode:
    def __init__(self, key, value):
        self.left  = None
        self.right = None
        self.color = BLACK
        # pola parent nie ma -- nie jest w tym zadaniu potrzebne
        self.key   = key
        self.value = value
        # WOLNO dopisać swoje pola
```

Proszę zaimplementować funkcję `checkRB(T)`, która dostaje na wejściu węzeł drzewa i sprawdza, czy zakorzenione w nim drzewo spełnia wszystkie warunki bycia drzewem czerwono-czarnym. Państwa kod będzie uruchamiany.