

WAŻNE1: Do zadania przygotuj 20 testów (komend) sprawdzających poprawność działania! (brak testów - 0 punktów)

WAŻNE2: Wolno używać tylko funkcji bibliotecznych o złożoności obliczeniowej $O(1)$.

1) Napisz moduł realizujący operacje na binarnym drzewie poszukiwań. Moduł powinien udostępniać na zewnątrz funkcje tylko w postaci sygnatury. Operacje do implementacji:

- `create()` - tworzy strukturę drzewa
- `push(Int)` - dodaje element do drzewa poszukiwań
- `remove(Int)` - usuwa element z drzewa poszukiwań
- `find(Int)` - sprawdź czy istnieje dany element
- `getPreOrder()` - pobiera listę leniwą przechodzącą przez wierzchołki w sposób PreOrder
- `getPostOrder()` - jw. tylko PostOrder
- `getInOrder()` - jw. tylko InOrder

Przykład:

```
T.create()
T.push(5)
T.push(3)
T.push(1)
T.push(4)
T.push(7)
```

```
T.getPreOrder() -> [5;3;1;4;7]
T.getPostOrder() -> [1;4;3;7;5]
T.getInOrder() -> [1;3;4;5;7]
```

Język: Ocaml