

# **Algorytmy i struktury danych, Teleinformatyka, I rok**

## **Raport z laboratorium nr: 6**

**Imię i nazwisko studenta: Paweł Młodkowski**

**nr indeksu: 418087**

*1. W pole poniżej wklej najważniejszy (według Ciebie) fragment kodu źródłowego z zajęć (maksymalnie 15 linii).*

```
def search(self, value):
    current = self
    if current.is_leaf() or current.value == value:
        return current
    if current.left is not None and value < current.value:
        return current.left.search(value)
    elif current.right is not None:
        return current.right.search(value)
```

*Uzasadnij swój wybór.*

Wybrałem algorytm wyszukiwający daną wartość w BST, bo jest zaskakująco prosty w zrozumieniu i implementacji, ale jednocześnie szybki.

*2. Podsumuj wyniki uzyskane podczas wykonywania ćwiczenia. Co ciekawego zauważyłeś? Czego się nauczyłeś? Jeśli instrukcja zawierała pytania, odpowiedz na nie. Do sprawozdania możesz dodać wykresy jeśli jest taka potrzeba.*

Zdecydowałem się zaimplementować strukturę obiektowo, tworząc klasę reprezentującą pojedynczy węzeł drzewa. To podejście z pewnością pomogło w intuicyjności, ale powodowało również problemy w implementacji niektórych algorytmów (patrzę szczególnie na insert, który w końcu zrobiłem jako funkcję, nie metodę klasy), wyświetlanie struktury również okazało się zaskakująco trudne ze względu na konieczność przenoszenia spacji z wywołania na wywołanie.

Sama struktura może znaleźć zastosowanie w przypadkach, kiedy jest dużo skupionych blisko siebie danych, bo to minimalizuje najgorzej zoptymalizowaną część, czyli tablicę zawierającą korzenie. Dodatkowo przeszukiwanie i dodawanie nowych danych do struktury jest bardzo szybkie w takim przypadku. Nie mam żadnego pomysłu na „use-case”, który pasowałby idealnie, ale myślę, że struktura mogłaby się sprawdzić w bazie ocen uczniów. (choć o wiele łatwiej i użyteczniej byłoby użyć czegoś jak SQL)

Jeżeli poprawnie zaimplementowałem funkcje z zad. 2, to:

- Złożoności obliczeniowe `search()`, `maximum()` i `minimum()` powinny liniowo zależeć od głębokości drzewa.
- Złożoność `insert()` powinna zależeć od zapełnienia drzewa (choć nie jestem pewny w jaki sposób)