

Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: a02

Imię i nazwisko studenta: Maciej Klimek

nr indeksu: 414836

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

```
def printTree(self, depth=0):
    print((depth) * "-" + f"{self.data}", end="")
    if len(str(self.data)) < 4:
        print(" ", end="")
    depth += 1
    if self.left != None:
        self.left.printTree(depth)
    else:
        print("")
    if self.right != None:
        padding = 0
        for i in range(depth):
            padding += i + 4
        print((padding) * " ", end="")
        self.right.printTree(depth)
```

Uzasadnij swój wybór.

Ta metoda printująca strukturę w drzewie binarnym jest wg mnie najciekawszą częścią tego ćwiczenia. Sporym problemem było dla mnie wymyślenie jak odpowiednio dobrać ilość znaków spacji przed elementem z kolejnej „prawej” gałęzi drzewa, tak by elementy na tym samym poziomie znajdowały się w tych samych kolumnach (co gorsza żeby same kolumny nie „rozjeżdżały się” wertykalnie). Rozwiązaniem które końcowo zaimplementowałem był for loop zależny od zmiennej depth, określającej ilość pauz (poziom na którym dany element się znajduje), każdego z elementów.

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.

Dane podane poniżej przeanalizowałem poprzed przedstawienie ich na wykresie.

Złożoność metody SEARCH jest całkiem bliska logarytmicznej - $O(\log(n))$

INSERT też jest całkiem bliski złożoności $O(\log(n))$, jednak nie tak dokładnie jak SEARCH

Dla metod MINIMUM i MAXIMUM ciężko oszacować jest złożoność obliczeniową przez zbyt małe różnice w zależności od ilości elementów

(dane na drugiej stronie)

Średni czas dla 25 elementów

INSERT: 5.767345428466797e-06

MINIMUM: 2.5463104248046874e-06

MAXIMUM: 2.379417419433594e-06

SEARCH: 9.62972640991211e-06

Średni czas dla 100 elementów

INSERT: 6.3158321380615234e-06

MINIMUM: 2.847381591796875e-06

MAXIMUM: 3.99725341796875e-06

SEARCH: 2.7587413787841795e-05

Średni czas dla 50 elementów

INSERT: 7.770061492919923e-06

MINIMUM: 3.7837028503417967e-06

MAXIMUM: 3.247261047363281e-06

SEARCH: 1.463174819946289e-05

Średni czas dla 1000 elementów

INSERT: 5.4502487182617186e-05

MINIMUM: 4.658367156982422e-05

MAXIMUM: 1.969337463378906e-05

SEARCH: 0.0001591205596923828