Algorytmy i struktury danych

Labolatorium 3 -Kopce

Skład zespołu:

* Jan Kwiatkowski
* Jarosław Nachyła

W ramach zajęć zaimplementowano kopiec n-arny.  
Algorytm kopca ***heap.py*** został stworzony **wspólnie**, natomiast wyświetlaniem kopca w algorytmie ***printer.py*** zajął się **Jan Kwiatkowski**, a testowaniem algorytmu w pliku ***experiments.py*** ***Jarosław Nachyła*.**

Plik ***heap.py –*** zawiera algorytm kopca.

Plik ***printer.py*** – zawiera algorytm rysowania kopca.

Plik ***experiments.py –***zawiera algorytm pomiaru czasu dodawania do 10 kopców 10 list każda zawierająca kolejno 10000,20000,...,100000 losowych wartości z zakresu 1 – 300000. Zawiera także pomiar czasu usuwania wszystkich elementów z kopców.

Zmierzone czasy dodawania i usuwania elementów z kopca porównano na wykresie przedstawionym poniżej:

Chart, line chart

Description automatically generated

Złożoności poszczególnych operacji kopca binarnego (oraz n-arnego), dane wzięte z prezentacji z wykładu:

* Największy (najmniejszy) element = 1
* Pop= O(logn)
* Push = O(logn)

Dla kolejnych rzędów kopca operacje dodawania i usuwania są coraz szybsze, wynika to z faktu, iż wraz ze wzrostem rzędu kopca zmniejsza się jego wysokość.

Poprawność działania kopca 2/3/4-arnego stworzonego z listy: [2,3,9,4,1,8,7,15,20,41,32,5,4,1,5,4,1,8,6,13,22,16,56,17] sprawdzono wyświetlając go w konsoli.

* 2-arny

A screenshot of a computer

Description automatically generated with medium confidence

* 3-arny

A screenshot of a computer

Description automatically generated with medium confidence

* 4-arny

