

AISDI – Kopce

Struktura projektu:

heaps.py – implementacja kopca

test_heaps.py – testy jednostkowe

benchmark.py – pomiary wydajności

ploter.py – tworzenie wykresów

Zaimplementowana została klasa reprezentująca kopiec. Kopiec o danej „krotności” tworzymy podając w konstruktorze parametr n . Dzięki temu przy użyciu jednej klasy zaimplementowaliśmy jednocześnie wszystkie wymagane rodzaje kopców.

Wyświetlanie kopca dla liczb z zakresu $\langle 1, 20 \rangle$:

Binarny:

```
#0: 20
#1: 19, 14
#2: 17, 18, 11, 13
#3: 10, 16, 9, 8, 2, 6, 5, 12
#4: 1, 7, 4, 15, 3, -, -, -, -, -, -, -, -, -, -, -
```

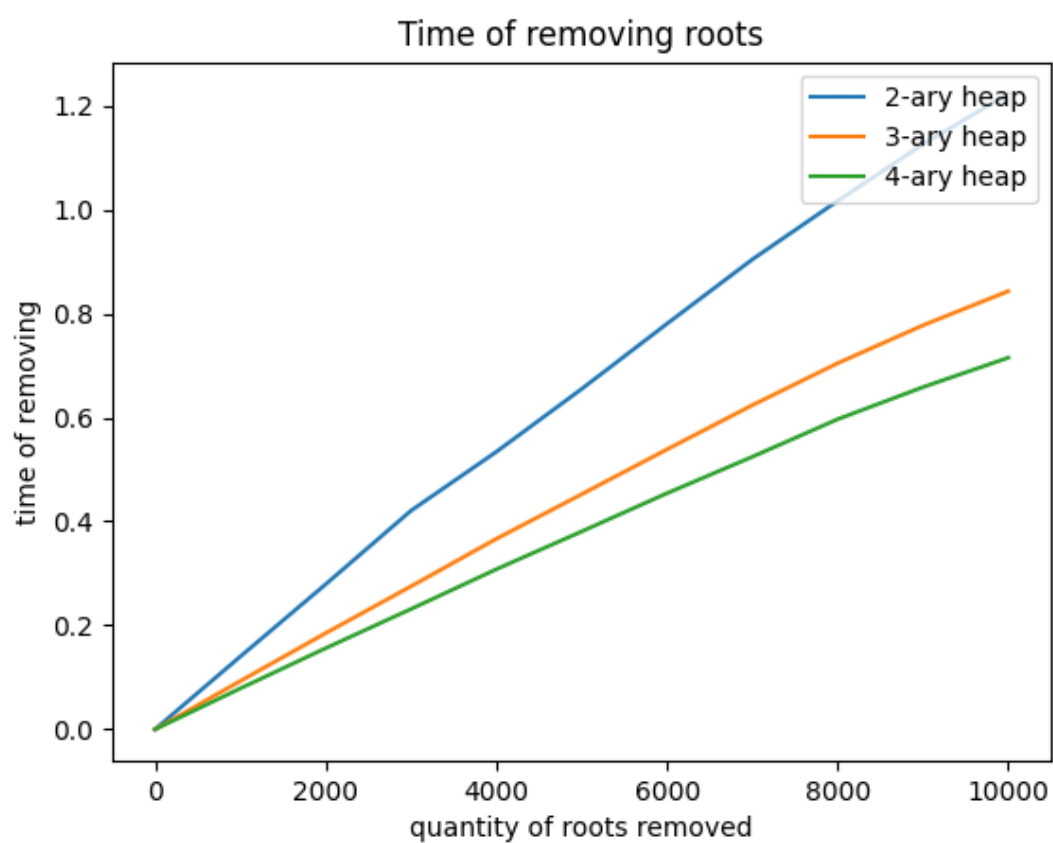
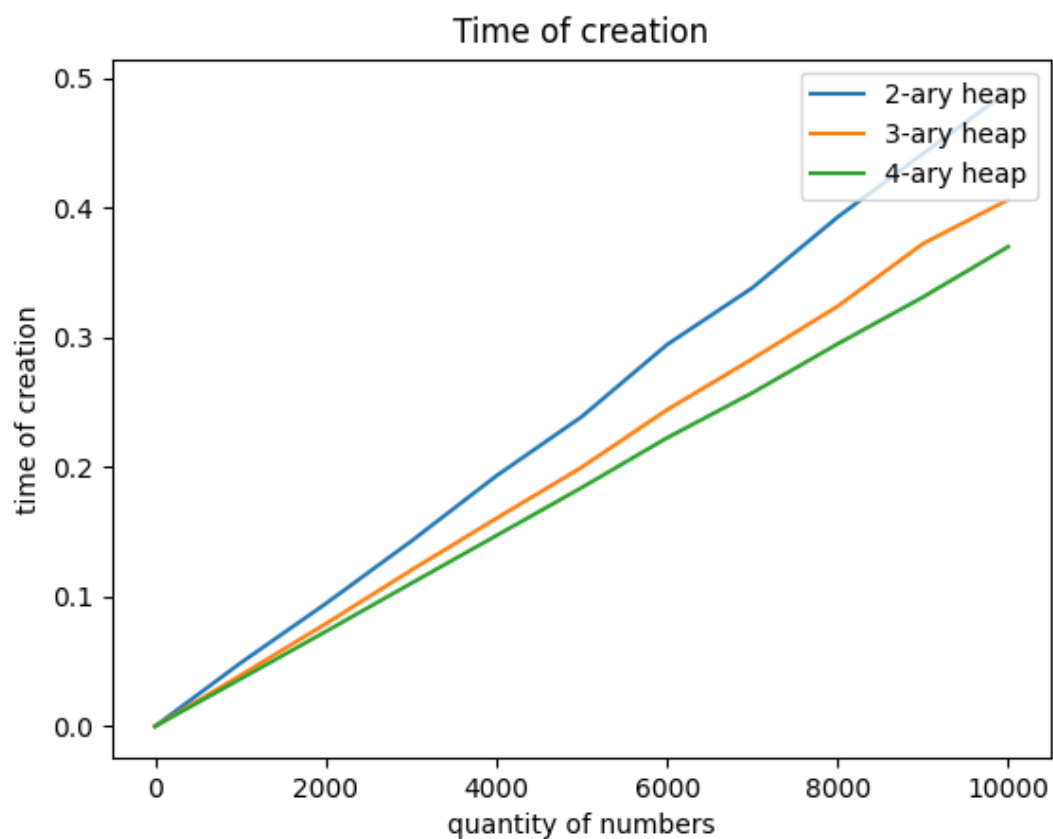
3-narny:

```
#0: 20
#1: 19, 9, 12
#2: 14, 17, 18, 2, 7, 8, 3, 10, 11
#3: 1, 6, 13, 4, 15, 16, 5, -, -, -, -, -, -, -, -, -, -, -, -
```

4-narny:

```
#0: 20
#1: 8, 12, 16, 19
#2: 1, 5, 6, 7, 2, 9, 10, 11, 3, 13, 14, 15, 4, 17, 18, -
```

Pomiary wydajności:



Widzimy, że w obu przypadkach najszybsze są operacje wykonywane na kopcu 4-narym, a najwolniejsze na kopcu binarnym. Wykresy są zbliżone do wykresów funkcji $n \log_i(n)$ dla i równego 2, 3, 4. Pokazuje to, że kopce o wyższej podstawie logarytmu będą działały szybciej.