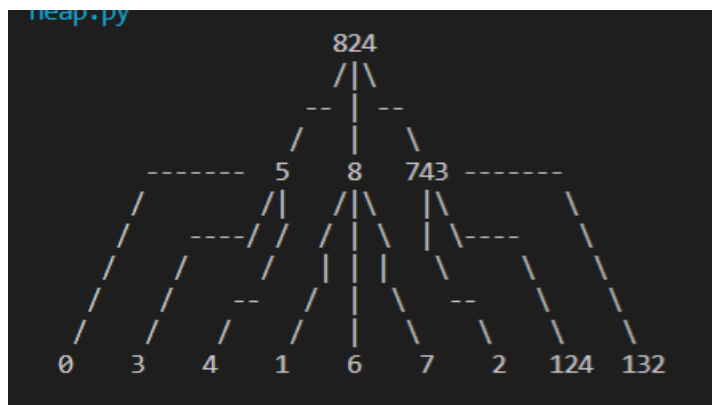
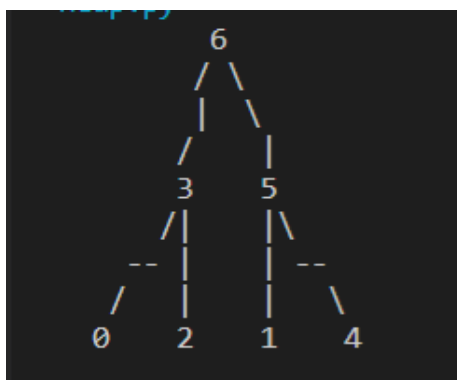


Kopiec zupełny to to drzewo binarne, w którym wartość każdego węzła poddrzewa jest większa lub równa wartościom elementów odchodzących od tego węzła. Kopce 2-arne mają strukturę standardowego drzewa, czyli każde poddrzewo składa się z węzła i dwóch odchodzących od niego elementów. Odpowiednio, dla kopców 3-arnych od węzłów odchodzą 3 elementy, a dla kopców 4-arnych, cztery. Elementy kopców 2-arnych umieszczane są w tablicy według następującego wzorca: --> Jeśli węzeł jest przechowywany pod indeksem k , to jego lewy element jest przechowywany pod indeksem $2k + 1$, a jego prawy element pod indeksem $2k + 2$. Dla kopców zupełnych 3-arnych: --> Jeśli węzeł jest przechowywany pod indeksem k , to jego lewy element jest przechowywany pod indeksem $3k + 1$, środkowy pod $3k + 2$, a jego prawy element pod indeksem $3k + 3$. Dla kopców zupełnych 4-arnych --> Jeśli węzeł jest przechowywany pod indeksem k , to jego najbardziej lewy element jest przechowywany pod indeksem $4k + 1$, drugi pod $4k + 2$, trzeci pod $4k + 3$, a jego najbardziej prawy element pod indeksem $4k + 4$. Napisałyśmy jedną uniwersalną klasę reprezentującą różne kopce. Kopiec przyjmuje jako argument konstruktora informację ile dzieci ma mieć każdy rodzic.

Umieszczanie nowego elementu – wstawiany jest na pierwsze wolne miejsce i następnie porównywany ze swoim rodzicem. Jeżeli rodzic jest mniejszy od nowego elementu - zostają zamienione miejscami. Następnie porównywany jest ze swoim nowym rodzicem i tak do momentu gdy rodzic będzie większy od nowego elementu lub dojdzie on do korzenia.

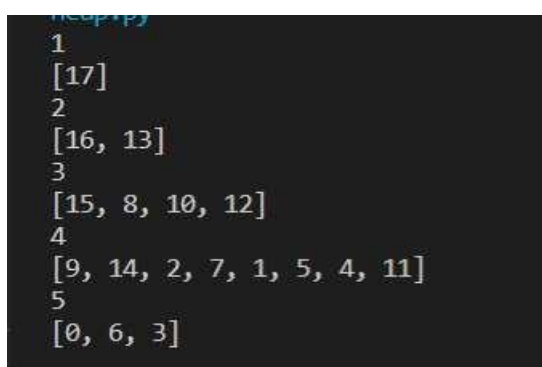
Usuwanie korzenia – na miejsce korzenia wstawiany jest ostatni element kopca, a następnie nowy korzeń porównywany jest ze swoimi dziećmi i jeżeli którekolwiek z dzieci jest on niego większe zamieniany jest z największym z nich. Następnie porównywany jest ze swoimi nowymi dziećmi i tak do momentu gdy będzie większy od wszystkich dzieci lub nie będzie miał żadnych dzieci.

Wypisywanie – zapewniłyśmy 2 możliwości wypisywania. W formie graficznej drzewa dostępne jest tylko dla kopca 2-arnego i 3-arnego tylko do maksymalnie 3 poziomów i elementów maksymalnie 3 cyfrowych:



Ponieważ przy innych warunkach stawało by się nieczytelne. Wtedy wyświetlane jest są osobno kolejne poziomy kopca:

2-arny



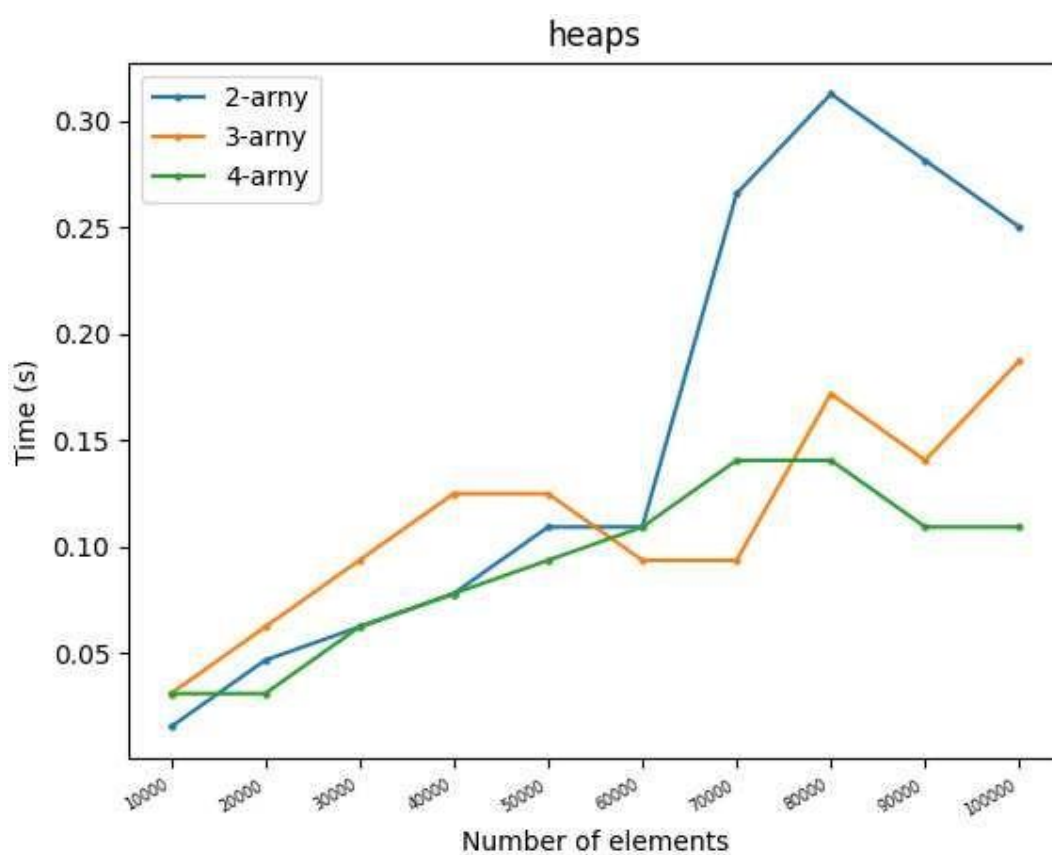
3-arny

```
1
[824]
2
[743, 29, 38]
3
[132, 16, 19, 22, 25, 28, 31, 34, 37]
4
[39, 124, 12, 3, 14, 15, 4, 17, 18, 1, 8, 21, 6, 23, 24, 7, 26, 27, 2, 11, 30, 9, 32, 33, 10, 35, 36]
5
[0, 13, 20, 5]
```

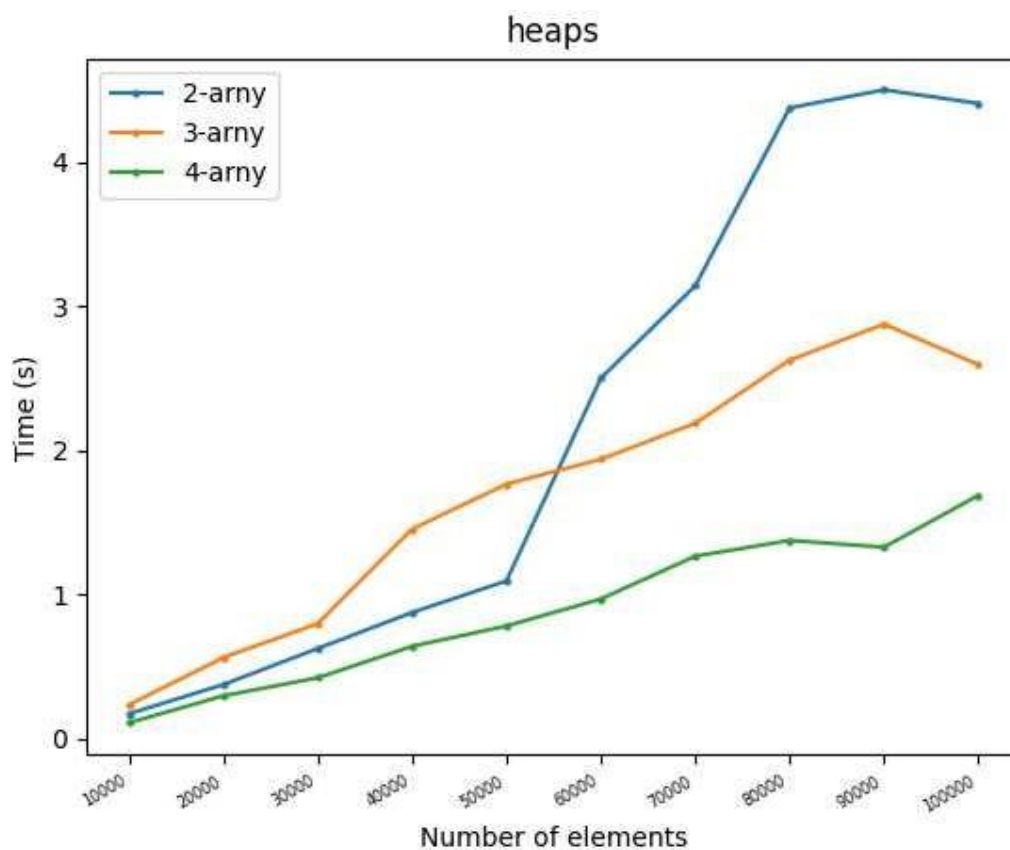
4-arny

```
1
[824]
2
[35, 51, 743, 19]
3
[22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 132, 3, 16, 17, 18]
4
[0, 7, 20, 21, 4, 23, 24, 25, 5, 27, 28, 29, 6, 31, 32, 33, 1, 11, 36, 37, 8, 39, 40, 41, 9, 43, 44, 45, 10, 47, 4
8, 49, 2, 15, 52, 53, 12, 55, 56, 57, 13, 59, 60, 61, 14, 124]
```

Czasy wstawiania nowych elementów do kopca prezentują się następująco:



Czasy usuwania największych elementów:



Ze względu na stosunkowo krótkie czasy przy wstawianiu nie wychodzi jednoznacznie, że kopiec 4-arny jest szybszy od 3-arnego a on od 2-arnego jak powinno to wyglądać książkowo. Prawdopodobnie przy większych danych różnica ta byłaby bardziej widoczna. Lepiej widoczne jest to przy usuwaniu. Złożoność wstawiania dla kopca x-arnego wynosi $\log_x(n)$

