**Algorytmy i Struktury Danych  
Projekt 1  
„**Symulacja Systemów”

1. **Kolejka Priorytetowa**Interfejs kolejki priorytetowej zawiera 2 metody:  
   **1. void dodaj(D obiekt, K klucz)**

- metoda służąca do dodawania elementów do kolejki priorytetowej.Parametrami wejściowymi są obiekt D dane oraz K klucz (obiekty generyczne, użytkownik nadaje typ tym elementom).

**2. Element<D,K> usun()**

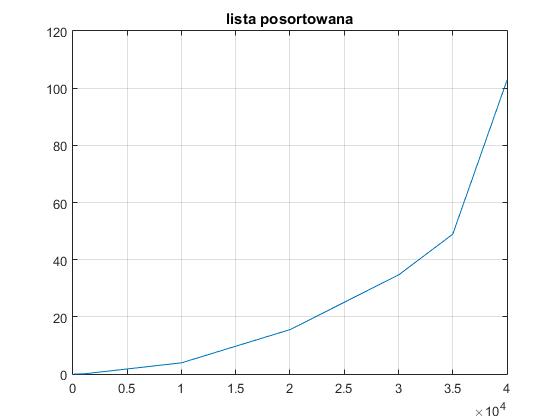
- metoda służąca do usuwania elementów z kolejki priorytetowej. Usuwa element o największej wartości klucza. Obiekt zwracany jest całościowo (zarówno dane, jak i klucz).

Kolejka została zaimplementowana przez dwie klasy dziedziczące z powyższego interfejsu:

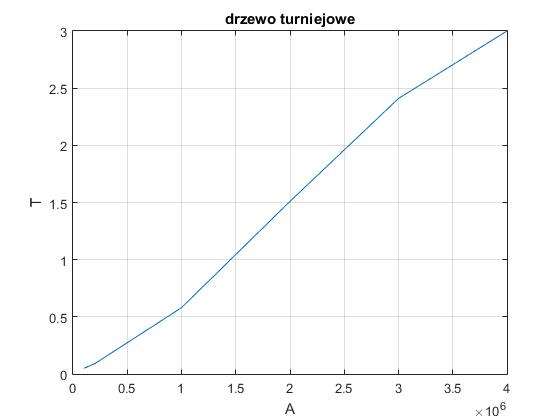
1. public class **ListaPosortowana<D,K>**
2. public class **DrzewoTurniejowe<D,K>**

Pierwszym sposobem implementacji było zastosowanie listy uporządkowanej. Lista jest uporządkowana rosnąco, element o największej wartości klucza znajduję się na końcu listy.

Drugim sposobem implementacji jest struktura drzewiasta, a mianowicie drzewo turniejowe.  
Największy element drzewa (o największym kluczu, czyli zwycięzca wszystkich pojedynków) znajduję w korzeniu.

Metodę testującą dla pierwszego sposobu wywołano dla liczby wstawień A należącej do zakresu [0; 40000], maksymalnych wartościach kluczy M = 1000. 

Natomiast druga metoda została przetestowana dla wartości A należących do przedziału [1000;4000000] dla takich samych wartości klucza M.



Wykresy prezentują zależności czasu od liczby wstawianych elementów. Złożoność obliczeniowa dla struktury drzewiastej jest znacznie lepsza, jak widać na załączonych wykresach, teoretycznie wynosi ona O(logn), natomiast lista uporządkowana ma złożoność obliczeniową w przybliżeniu liniową, co niestety nie jest dobrze przedstawione na danym wykresie, ponieważ wartości obliczeniowe dla większej ilości elementów stają się zauważalnie przeszkadzające.