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).

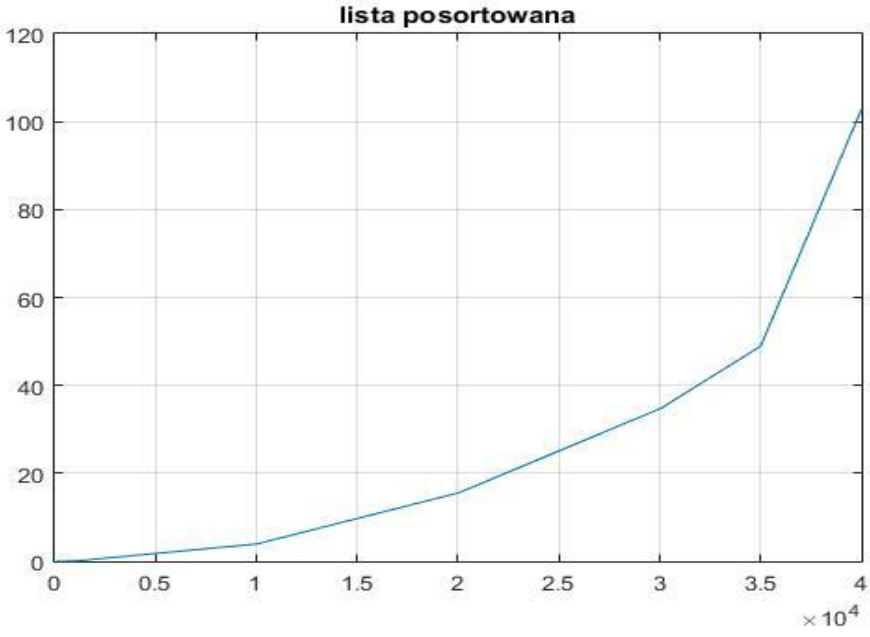
Kolejka została zaimplementowana przez dwie klasy dziedziczące z powyższego interfejsu:

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

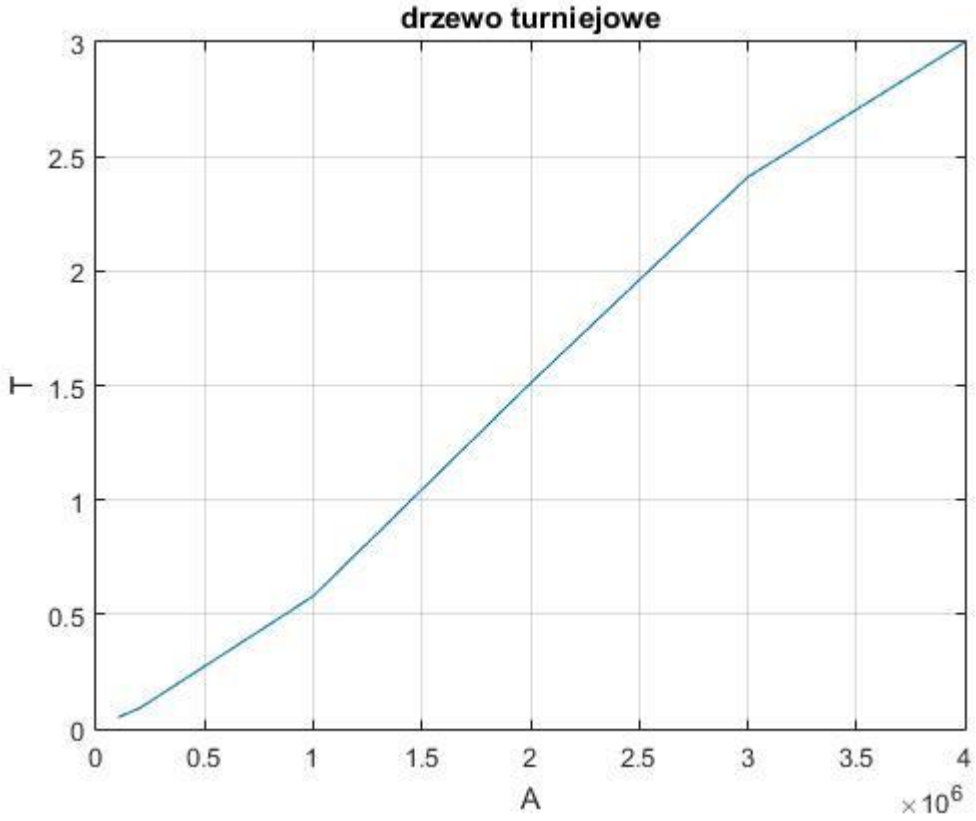
Pierwszym sposobem implementacji było zastosowanie listy uporządkowanej. Lista jest uporządkowana rosnąco, element o największej wartości klucza znajduję się na końcu listy.

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.

2. Symulacja Centrali Telefonicznej

Klasy zawarte w centrali:

- **Class Fifo** – implementacja kolejki fifo w buforze centrali. Jest odpowiedzialna za przetrzymywanie zgłoszeń nie mogących dostać się na wiązkę wyjściową z powodu braku miejsca. Zgłoszenia są z niej usuwane w przypadku, gdy maksymalny czas oczekiwania danego zgłoszenia jest mniejszy od czasu skoku. Elementami kolejki jest tablica zgłoszeń **private Zgloszenie[] element**, zmienna mówiąca nam o aktualnej pozycji w kolejce **private int wskaźnik** oraz zmienna mówiąca o wielkości kolejki, **private int pojemność**.
- **Class Rozklad** – klasa służąca do przeliczenia wartości danych określonych wykładniczymi rozkładami prawdopodobieństwa.
- **Class Statystyki** – klasa pomocnicza, służąca do wyliczania statystyk systemu.

- **Class Strumien** – klasa definiująca strumienie wchodzące do centrali. Każdy strumień nadaje wartości nadchodzącemu zgłoszeniu.
- **Class Zgloszenie** – klasa opisująca nadchodzące zgłoszenie.
- **Class Symulacja** – główna klasa projektu, zawierająca metody wczytywania oraz zapisu do pliku, a także metodę samej symulacji.

Przebieg symulacji:

Symulację systemu zaczynamy od wczytania danych z pliku, wczytania czasu symulacji z klawiatury, zainicjowanie listy zgłoszeń oraz klas pomocniczych, takich jak Rozklad, Strumien, Zgloszenie i Statystyki. Dla każdego strumienia następuje powiązanie jego pól zależnych od rozkładu ekspotencjalnego za pomocą metody przypisanieRozkladow(). Następnie inicjujemy przyjście zgłoszenia do strumienia, sprawdzając jednocześnie, czy owy strumień jest w stanie przyjąć to zgłoszenie (czy czas odstępu między zgłoszeniami minął). Jeśli czas ten minął, sprawdzamy zawartość kolejki systemu. Gdy jest ona pusta, wrzucamy zgłoszenie na wiązkę kanałów w systemie, gdy są w niej jakieś elementy, ale nie jest pełna to dodajemy dane zgłoszenie. Ostatnim przypadkiem jest zapełnienie danej kolejki co implikuje brak miejsca dla kolejnego zgłoszenia (zgłoszenie zostaje odrzucone). Po nadejściu danego zgłoszenia nadajemy czas odstępu do nadejścia kolejnego zgłoszenia. Zawartością danego zgłoszenia są dane dotyczące czasu przebywania w systemie oraz maksymalnego czasu oczekiwania (przebywania w kolejce systemu). Tak więc następnym krokiem jest ustalenie skoku czasowego. Robimy to poprzez porównanie ze sobą czasów odstępu na strumieniach oraz porównanie czasów na wiązce systemowej (trwania połączenia). Następnym krokiem jest porównanie najkrótszych czasów z wcześniej porównywanych. Po czym od wszystkich czasów(w kolejce, odstępu strumieni oraz w systemie) odejmujemy ten najmniejszy czas. Podczas tych procedur pamiętamy oczywiście o możliwości zbyt długiego oczekiwania w kolejce. Każdy skok jest porównywany z czasem naszej symulacji. Na koniec obliczamy statystyki systemowe i wypisuje je do pliku txt.