**Maciej Oziębły 184147**

**Tomasz Wojciechowski 184132**

Sterowanie procesami dyskretnymi

Sprawozdanie 3

Zakres sprawozdania:

* algorytm Shrage
* algorytm Shrage z podziałem zadań
* algorytm Caliera

## Środowisko

System operacyjny: Windows 7   
Platforma programistyczna: .NET 4.5  
Język programowania: C#  
IDE: Visual Studio 2012 Ultimate (wersja z MSDNAA)  
Komputer wypozażony w 2 rdzeniowy procesor korzystający z technologii Hyper-threading (więc mogący wykonywać równolegle 4 operacje)

## Implementacja

Implementacja algorytmów, oprogramowania pomocniczego i optogramowania testującego poprawność algorytmów oraz oprogramowania testującego szybkość działania algorytmów jest dostępna w plikach:

* Calier.cs – implementacja algorytmu Caliera
* Heap.cs – generyczna implementacja stosu
* Helpers.cs – funkcje pomocnicze
* OrderingTests.cs – testy funkcji testujących
* PrmtShrage.cs – implementacja algorytmu Shrage z przerwaniami
* Progams.cs – funkcja main
* Shrage.cs – implementacja algorytmu Shrage z kopcem i bez kopca
* Task.cs – implementacja klasy pomocniczej „Task”

## Poprawność implementacji algorytmów

W celu ciągłego sprawdzania działania poprawności działania implementacji algorytmów stworzyliśmy testy automatyczne dla wszystkich implementacji, które porównują otrzymane wyniki z wynikami podanymi wraz z zadaniem.

W teh chwili wszystkie algorytmy przechodzą te testy, a do tego przechodzą testy testerką dostarczoną z zadaniem.

## Wydajność implementacji algorytmów

Wykonaliśmy po 500 prób wykonania algorytmu dla każdego z plików i otrzymaliśmy następujące średnie czasy wykonania:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| wielkość instancji | Shrage z kopcem[ms] | Shrage bez kopca[ms] | Shrage z przerwaniami[ms] | Calier[ms] |
| 50 | 0,084 | 0,156 | 0,078 | 0,622 |
| 100 | 0,102 | 0,394 | 0,098 | 1,756 |
| 200 | 0,216 | 1,372 | 0,234 | 17,294 |

Z testów wydajności bardzo ciężko określić złożoność obliczeniową algorytmów. Przy przejściu z wielkości instancji 50 na 100 zmiana czasu działania algorytmów jest niewielka, natomiast różnica przy przejściu z instancji 100 na 200 jest ogromna. Spowodowane jest to prawodopodobnie budową tych instancji – pokazuje to że czas wykonania tych algorytmów nie jest zależny tylko od wielkości instancji, ale też od tego z jakich zadań się one składają.

Wydajność algorytmu Shrage bez użucia kopca (z użyciem vectora i sortowania algorytmem „intrasort” jest o wiele gorsza niż wydajność algorytmu z użyciem kopca). Wiąże się to ze zwiększeniem złożoności obliczeniowej pewnych operacji przy przeszukiwaniu kontenera przez co rośnie złożoność obliczeniowa całego algorytmu (co doskonale widać na wynikach w tabeli).