

# Sprawozdanie z laboratorium 3

Filip Malinowski

26 marca 2015

W zadaniu są dwa programy. Pierwszy generuje dane wejściowe. Drugi jest odpowiedzialny za uruchamianie wybranych algorytmów. Oba programy uruchamiane są przez Makefile. W programie zostały wprowadzone poprawki oraz możliwość zwracania wartości usuwanych elementów ze struktur przechowujących dane. Ilość danych wejściowych jest konfigurowana w kodzie źródłowym w pliku benchmark.hh. Program testujący algorytmy składa się z klasy bazowej Benchmark i klas pochodnych:

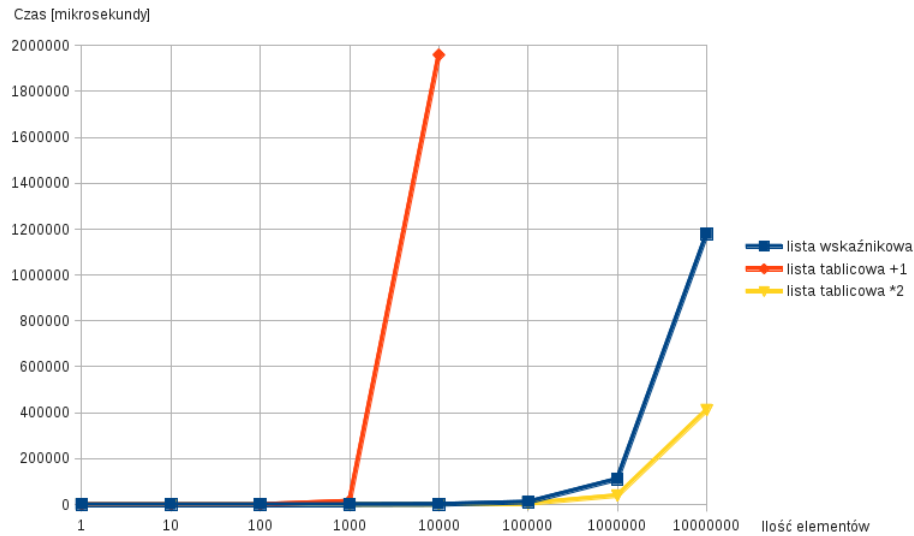
- AlgorithmStos
- AlgorithmKolejka
- AlgorithmLista
- Algorithm2

Klasa Benchmark wyznacza odstęp czasowy za pomocą pomiaru czasu przy użyciu biblioteki chrono. Następnie tworzony jest plik wyjściowy o wybranej nazwie, do którego zapisywane są dane o szybkości wykonania algorytmu. Wygenerowane dane to:

1. wprowadzanie do listy tablicowej z inkrementacją rozmiaru o 1 - tab list plus 1.txt
2. wprowadzanie do listy tablicowej z pomnażaniem rozmiaru o 2 - tab list multip 2.txt
3. wprowadzanie do listy z powiększaniem o 1 - tab plus 1.txt

Na załączonym wykresie można zauważyć, że najgorszy czas działania ma wprowadzanie do listy tablicowej z inkrementacją rozmiaru o 1. Najlepszy czas działania ma lista tablicowa z podwajaniem rozmiaru.

Złożoność obliczeniowa wprowadzania elementów do listy tablicowej z inkrementacją o 1 zgadza się z przewidywaniami teoretycznymi. Wraz z dodawaniem elementów  $n$ -elementów, przy każdym dodaniu elementu lista jest powiększana. A powiększanie odbywa się przez przepisywanie całej zawartości tablicy przy każdym dodaniu. Zatem złożoność obliczeniowa powinna być zbliżona do  $n^2$ . Przy dodawaniu elementów do



Rysunek 1: Wykres przedstawiający czas wczytywania elementów w programie

listy tablicowej powiększanej dwukrotnie, złożoność obliczeniowa powinna być zbliżona do  $n * \log 2$ . Z otrzymanych danych można wywnioskować, że czas jest zbliżony do powyższych przewidywań. Czas dodawania elementów do listy jest wolniejszy od dodawania do listy tablicowej powiększanej dwukrotnie przez większą ilość wykonywanych operacji: tworzenie obiektu, a co za tym idzie wywołanie konstruktora, większej ilości warunków, przepisywanie wskaźników, itp. gdzie w przypadku listy tablicowej występuje jedynie powiększanie miejsca i przepisywanie elementu. W tym przypadku dla większych rozmiarów listy tablicowej metoda powiększania nie jest uruchamiana z powodu alokowania odpowiednich zapasów pamięci co przekłada się na szybkość działania.

Lista tablicowa +1		Lista tablicowa *2		Lista wskaźnikowa +1	
elem	time	elem	time	elem	time
1	1	1	1	1	2
10	9	10	6	10	1
100	166	100	7	100	12
1000	14851	1000	38	1000	118
10000	1959324	10000	483	10000	1095
		100000	4630	100000	10839
		1000000	39375	1000000	110985
		10000000	412031	10000000	1179047

Rysunek 2: Tabela z wartościami wypisanymi na wykresie