

Sprawozdanie z ćwiczenia laboratoryjnego

Tomasz Mikołajewski

2014

Spis treści

1	Wprowadzenie	1
2	Kod programu	2
3	Pomiary	2
4	Wyniki pomiarów	3
4.1	Stos stworzony na podstawie tablicy	3
4.1.1	Powiększanie tablicy za każdym razem	3
4.1.2	Powiększanie tablicy dwukrotnie	4
4.2	Kolejka stworzona na podstawie tablicy	5
4.2.1	Powiększanie tablicy za każdym razem	5
4.2.2	Powiększanie tablicy dwukrotnie	7
4.3	Stos stworzony na podstawie listy	8
4.4	Kolejka stworzona na podstawie listy	9
5	Wnioski	9

1 Wprowadzenie

Niniejszy dokument powstał w ramach przedmiotu Projektowanie Algorytmów i Metod Sztucznej Inteligencji. Jest on rezultatem przeprowadzonych ćwiczeń w laboratorium oraz pracy wykonanej w domu.

Ćwiczenie polegało na przeprowadzeniu analizy zapisu do struktur stosu oraz kolejki. Struktury zostały stworzone w oparciu o tablicę oraz listę. Podczas wykonywania czynności przepisywania liczony był czas wykonania ope-

racji. Testowanie powtarzano wielokrotnie, aby uzyskać ilość danych pozwalającą na wyznaczenie przybliżonej funkcji opisującej złożoność algorytmu.

2 Kod programu

Program został oparty o wcześniej stworzone pliki zawierające kod potrzebny do wykonania *benchmark-u* podprogramu, czyli określenia czasu jaki potrzebny jest na jego wykonanie. Główny program zawiera funkcję otwierającą pliki z danymi oraz zapisującą rezultat operacji do pliku. Zdecydowanie ułatwia to prace przy analizie zadanego algorytmu.

Struktury, które były testowane to:

- stos stworzony za pomocą tabeli
- stos stworzony za pomocą listy
- kolejka stworzona za pomocą tabeli
- kolejka stworzona za pomocą listy

Dodatkowo algorytmy zrealizowane na tablicy posiadały dwie możliwości kompilacji kodu: tablica powiększająca się za każdym razem oraz tablica powiększająca się razy 2.

3 Pomiary

Pomiary zostały przeprowadzone na dużych plikach zawierających zmienne typu *int*. Każdy z algorytmów był testowany dla co najmniej 5 różnych rozmiarach problemów powtórzonych dla dwóch różnych plików. Aby otrzymać rzetelny pomiar, każdy z testów był przeprowadzony 3-krotnie. Oznacza to iż każdy z algorytmów był testowany co najmniej 30 razy.

4 Wyniki pomiarów

Po przeprowadzeniu serii pomiarów otrzymano wyniki przedstawione w tabelach. Na podstawie wyników utworzono wykresy zamieszczone poniżej.

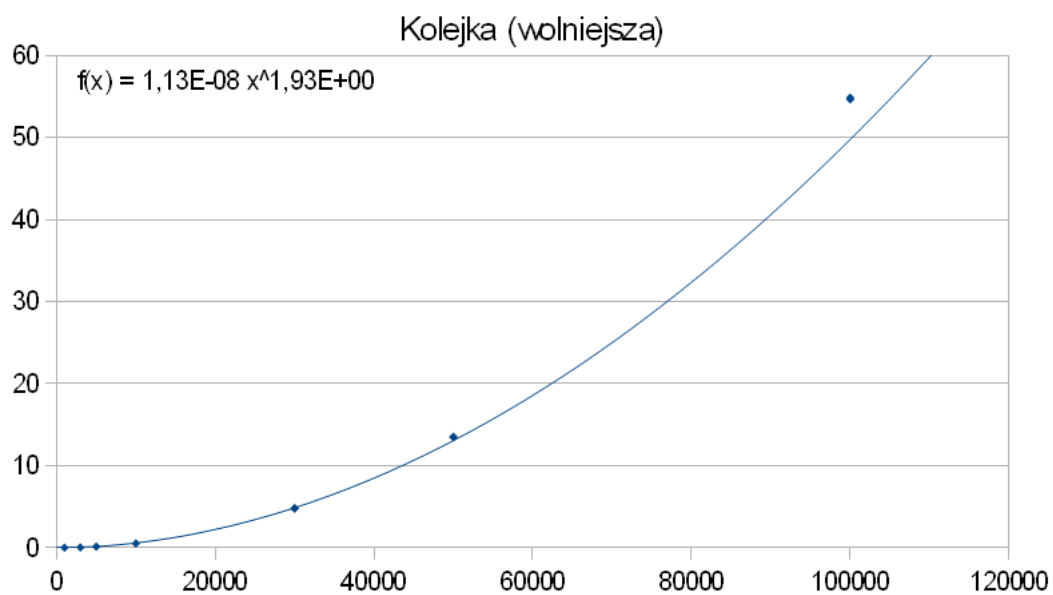
4.1 Stos stworzony na podstawie tablicy

Grupa ta zawiera dwie podgrupy ze względu na rodzaj wykonywanego algorytmu.

4.1.1 Powiększanie tablicy za każdym razem

Metoda wykonująca większą ilość obliczeń.

Wielkosc	1	2	3	Średnia
1000	0	0	0,015	0,005
1000	0	0,016	0	0,0053
3000	0,046	0,047	0,047	0,0467
3000	0,047	0,063	0,047	0,0523
5000	0,14	0,141	0,125	0,1353
5000	0,141	0,141	0,125	0,1357
10000	0,547	0,531	0,531	0,5363
10000	0,531	0,547	0,547	0,5417
30000	4,812	4,843	4,922	4,859
30000	4,812	4,828	5,078	4,906
50000	13,344	13,5	14,235	13,693
50000	13,469	13,781	14,015	13,755
100000	53,781	55,828	56,938	55,5157
100000	54,391	55,719	64,75	58,2867

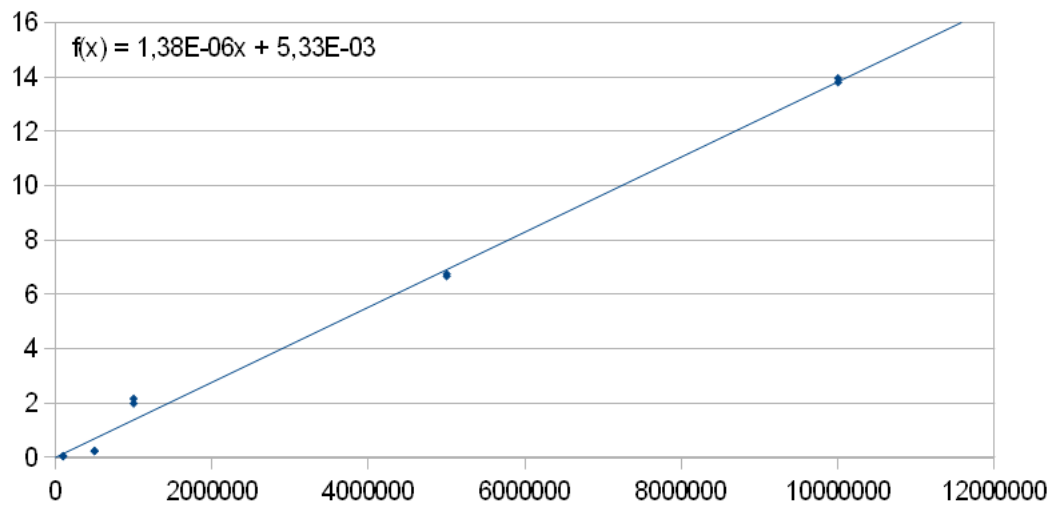


4.1.2 Powiększanie tablicy dwukrotnie

Metoda wykonująca mniejszą ilość obliczeń.

Wielkosc	1	2	3	Średnia
1100000	0,05	0,05	0,06	0,05
100000	0,05	0,06	0,05	0,05
500000	0,22	0,23	0,25	0,23
500000	0,23	0,24	0,25	0,24
1000000	2,03	2,14	2,08	2,08
1000000	1,91	2,3	2,33	2,18
5000000	6,42	6,67	6,81	6,64
5000000	6	6,98	6,92	6,64
10000000	13,09	13,88	14,36	13,78
10000000	13,02	13,97	14,03	13,67

Kolejka (szybsza)



4.2 Kolejka stworzona na podstawie tablicy

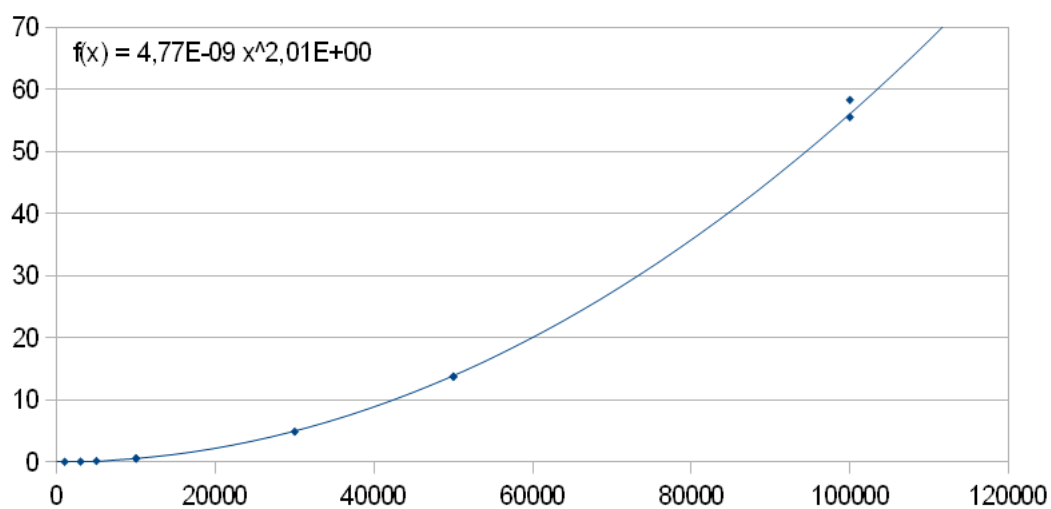
Grupa ta zawiera dwie podgrupy ze względu na rodzaj wykonywanego algorytmu.

4.2.1 Powiększanie tablicy za każdym razem

Metoda wykonująca większą ilość obliczeń.

Wielkosc	1	2	3	Średnia
1000	0	0	0,015	0,005
1000	0	0,016	0	0,0053
3000	0,046	0,047	0,047	0,0467
3000	0,047	0,063	0,047	0,0523
5000	0,14	0,141	0,125	0,1353
5000	0,141	0,141	0,125	0,1357
10000	0,547	0,531	0,531	0,5363
10000	0,531	0,547	0,547	0,5416
30000	4,812	4,843	4,922	4,859
30000	4,812	4,828	5,078	4,906
50000	13,344	13,5	14,235	13,693
50000	13,469	13,781	14,015	13,755
100000	53,781	55,828	56,938	55,5156
100000	54,391	55,719	64,75	58,2866

Stos (wolniejszy)

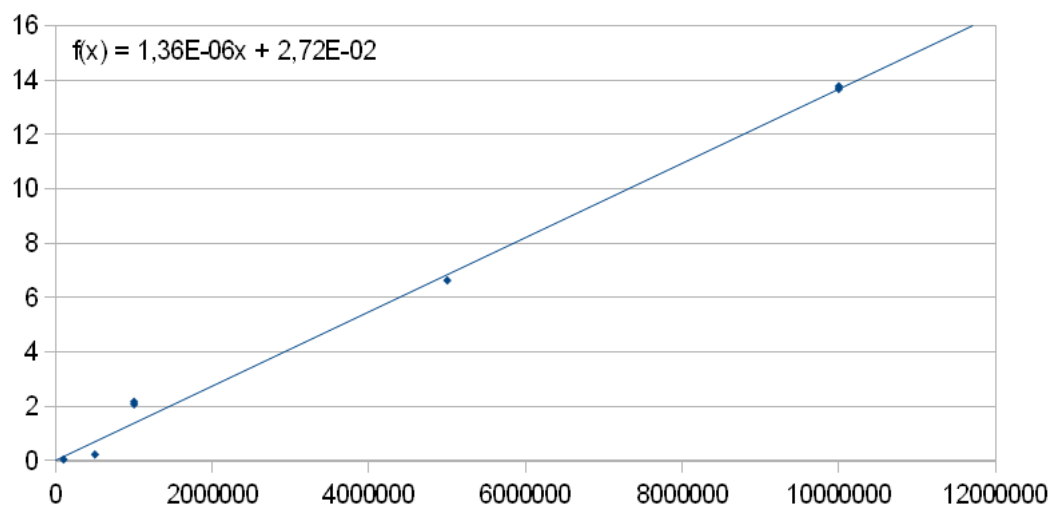


4.2.2 Powiększanie tablicy dwukrotnie

Metoda wykonująca mniejszą ilość obliczeń.

Wielkosc	1	2	3	Średnia
100000	0,047	0,047	0,062	0,052
100000	0,046	0,062	0,047	0,052
500000	0,219	0,234	0,25	0,234
500000	0,234	0,235	0,25	0,240
1000000	2,031	2,141	2,078	2,083
1000000	1,906	2,297	2,328	2,177
5000000	6,422	6,672	6,812	6,635
5000000	6	6,984	6,922	6,635
10000000	13,094	13,875	14,359	13,776
10000000	13,015	13,969	14,031	13,672

Stos (szybszy)

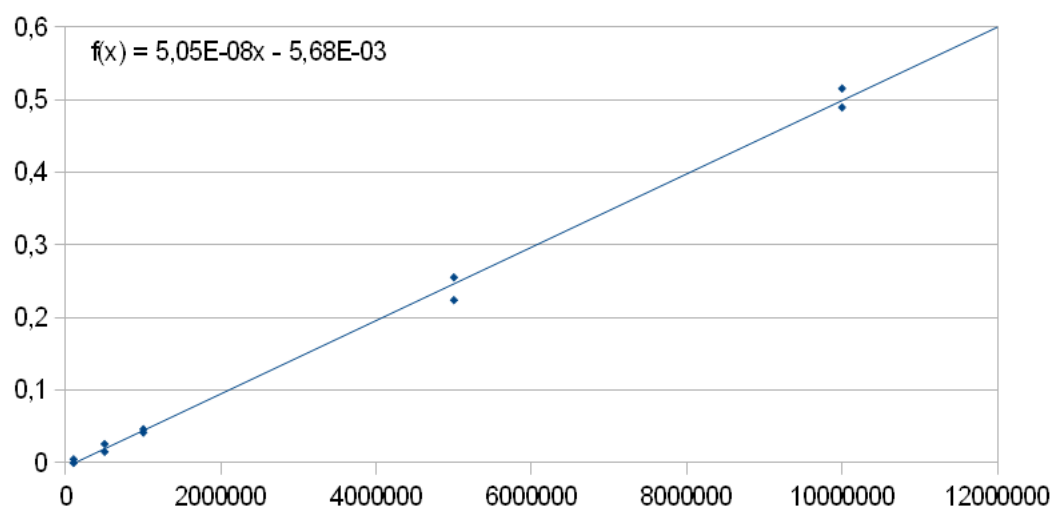


4.3 Stos stworzony na podstawie listy

Klasa odpowiadająca za przechowywanie danych stworzona na podstawie klasy *stack*.

Wielkosc	1	2	3	Średnia
100000	0	0	0	0
100000	0,015	0	0	0,005
500000	0,015	0,016	0,015	0,015
500000	0,031	0,016	0,031	0,026
1000000	0,047	0,031	0,047	0,042
1000000	0,062	0,046	0,031	0,046
5000000	0,219	0,203	0,344	0,255
5000000	0,219	0,219	0,234	0,224
10000000	0,671	0,391	0,484	0,515
10000000	0,515	0,5	0,453	0,489

Lista (stos)

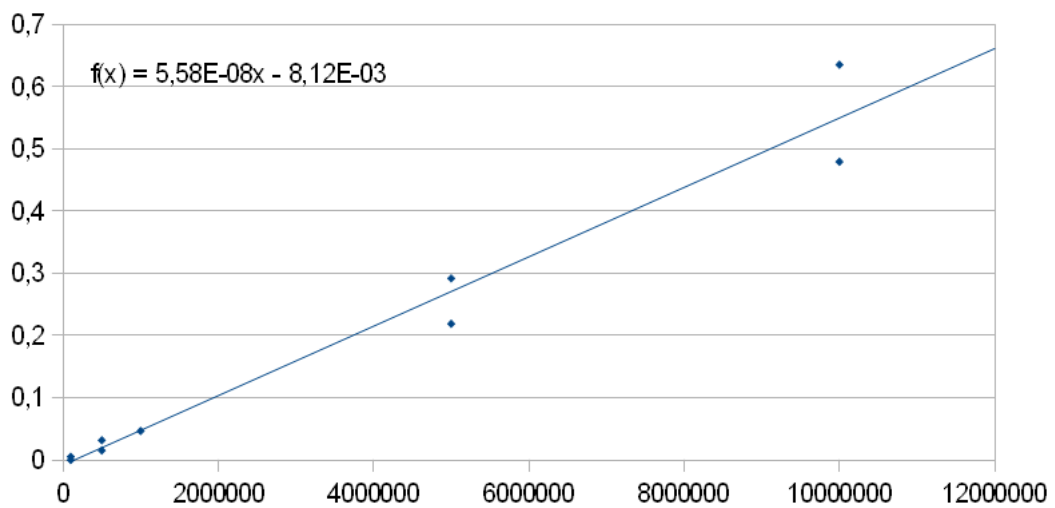


4.4 Kolejka stworzona na podstawie listy

Klasa odpowiadająca za przechowywanie danych stworzona na podstawie klasy *queue*.

Wielkosc	1	2	3	Średnia
100000	0	0	0,016	0,005
100000	0	0	0	0,000
500000	0,016	0,015	0,015	0,015
500000	0,032	0,032	0,031	0,032
1000000	0,031	0,031	0,078	0,047
1000000	0,062	0,031	0,047	0,047
5000000	0,203	0,235	0,219	0,219
5000000	0,25	0,235	0,391	0,292
10000000	0,454	0,547	0,438	0,480
10000000	0,485	0,469	0,953	0,636

Lista (kolejka)



5 Wnioski

Na podstawie danych i wykresów można wyciągnąć następujące wnioski:

- algorytmy przepisujące tablice za każdym razem mają złożoność obliczeniową o postaci wykładniczej
- algorytmy podwajające tablice mają złożoność obliczeniową zbliżoną do liniowej
- z testowanych struktur najszybsze okazały się listy
- złożoność obliczeniowa odgrywa duże znaczenie w skuteczności algorytmu
- różnice czasowe pomiędzy kolejką i stosem są niewielkie
- zgodnie z oczekiwaniami wartość liczb zapisanych w plikach testowych nie miała wpływu na otrzymywany czas (o ile były to liczby tego samego typu)