Projektowanie algorytmów i metody sztucznej inteligencji

Sprawozdanie z laboratorium

Temat:

Złożoność obliczeniowa wczytywania danych do stosu. Implementacja tablicowa oraz na liście.

1. Cel ćwiczenia

Celem ćwiczenia było porównanie złożoności obliczeniowej wczytywania danych do stosu w zależności od sposobu implementacji.

Zaimplementowany stos na tablicy dynamicznej należało przetestować dla różnych sposobów powiększania zaalokowanej tablicy.

Wyróżniliśmy 2 sposoby:

- gdy tablica zostanie przepełniona jej rozmiar powiększamy dwukrotnie, -gdy tablica zostanie przepełniona jej rozmiar powiększamy o jeden element.

2. Wyniki pomiarów

Wyniki pomiarów przedstawiam w postaci wykresu.

Wykres został stworzony w oparciu o dane z plików CSV, które zostały wygenerowane przez program benchmarkujący.

Dane wynikowe:

Zwiększanie tablicy dwukrotnie

Rozmiar	Ilość	Czas
problemu:	powtórzeń	wykonania
10	50	15.24
100	50	18.72
1000	50	51.16
10000	50	263.68
100000	50	2446.2

Zwiększanie tablicy o 1 element

Rozmiar	Ilość Czas	
problemu:	powtórzeń	wykonania
10	50	303.2
100	50	662.8
1000	50	7837.1
10000	50	350407



Z powyższego wykresu widać ze rozwiązania znacznie różnią się złożonością obliczeniową.

3.Wnioski

Złożoność obliczeniowa dwukrotnego zwiększania rozmiaru tablicy jest znacznie korzystniejsza w porównaniu z tą o 1 element. Wraz ze wzrostem wielkości problemu różnica jest coraz bardziej widoczna. Taki stan rzeczy spowodowany jest faktem, że gdy zwiększamy rozmiar o 1 element to musimy za każdym razem realokować pamięć na tablice. W drugim przypadku realokacja następuje znacznie rzadziej.

Laboratorium 3

Generated by Doxygen 1.8.1.2

Sun Mar 16 2014 23:33:05

Contents

1	PAM	ISI		1
2	Clas	s Index		3
	2.1	Class L	ist	3
3	Clas	s Docui	mentation	5
	3.1	Dzialar	nie Class Reference	5
		3.1.1	Detailed Description	5
		3.1.2	Member Function Documentation	5
			3.1.2.1 sprawdz	5
			3.1.2.2 wczytajDaneWejsciowe	6
			3.1.2.3 wlaczStoper	6
			3.1.2.4 wykonajAlgorytm	6
			3.1.2.5 wylaczStoper	6
	3.2	kolejka	_list Class Reference	7
		3.2.1	Detailed Description	7
		3.2.2	Member Function Documentation	7
			3.2.2.1 dequeue	7
			3.2.2.2 enqueue	7
			3.2.2.3 isempty	7
			3.2.2.4 size	8
	3.3	kolejka	_tab Class Reference	8
		3.3.1	Detailed Description	8
		3.3.2	Member Function Documentation	8
			3.3.2.1 dequeue	8
			3.3.2.2 enqueue	8
			3.3.2.3 isempty	9
			3.3.2.4 size	9
	3.4	Konten	er Class Reference	9
		3.4.1	Detailed Description	10
		3.4.2	Member Function Documentation	10
			2.4.2.1 dodaj olament	ın

ii CONTENTS

		3.4.2.2	dodaj_elementy	10
		3.4.2.3	operator+	10
		3.4.2.4	operator=	10
		3.4.2.5	operator==	11
		3.4.2.6	operator[]	11
		3.4.2.7	wczytajDane	11
		3.4.2.8	wez_dane	11
		3.4.2.9	wez_rozmiar	11
		3.4.2.10	zamien_elementy	12
3.5	stos_li	st Class Re	eference	12
	3.5.1	Detailed	Description	12
	3.5.2	Member	Function Documentation	12
		3.5.2.1	isempty	12
		3.5.2.2	pop	13
		3.5.2.3	push	13
		3.5.2.4	size	13
3.6	stos_ta	ab Class R	deference	13
	3.6.1	Detailed	Description	14
	3.6.2	Member	Function Documentation	14
		3.6.2.1	isempty	14
		3.6.2.2	pop	14
		3.6.2.3	push	14
		3.6.2.4	size	14
3.7	Tester	Class Refe	erence	14
	3.7.1	Detailed	Description	15
	3.7.2	Member	Function Documentation	15
		3.7.2.1	zamienNazwy	15

Chapter 1

PAMSI

2 **PAMSI**

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Dzialanie	
Klasa modelujaca gowna czesc programu	. 5
kolejka_list	
Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste	. 7
kolejka_tab	
Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste	. 8
Kontener	
Klasa Dane	. 9
stos_list	
Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste	. 12
stos_tab	
Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice	. 13
Tester	
Klasa Tester Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym po-	-
miarem	. 14

Class Index

Chapter 3

Class Documentation

3.1 Dzialanie Class Reference

Klasa modelujaca gowna czesc programu.

```
#include <dzialanie.hh>
```

Public Member Functions

void wczytajDaneWejsciowe (string nazwa, stos tab stos)

metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.

LARGE_INTEGER wlaczStoper ()

Metoda uruchamia pomiar czasu.

LARGE INTEGER wylaczStoper ()

Metoda konczy pomiar czasu.

void wykonajAlgorytm ()

Metoda wykonuje algorytm na danych wejsciowych (tablicy)

• bool sprawdz ()

Metoda sprawdza poprawnosc algorytmu.

• int uruchom (string nazwa)

Metoda wykonuje jednorazowy test algorytmu Metoda: -wlacza zegar -wykonuje algorytm -wylacza zegar -sprawdza poprawnosc algorytmu return czas wykonywania algorytmu w milisekundach.

• int wez_rozmiar ()

3.1.1 Detailed Description

Klasa modelujaca gowna czesc programu.

Klasa modeluje glowna czesc porgramu, ktorego zadaniem jest: -wczytanie danych -zmierzenie czasu dzialania algorytmu -sprawdzenie poprawności tego algorytmu, majac oczekiwany wynik

3.1.2 Member Function Documentation

```
3.1.2.1 bool Dzialanie::sprawdz ( )
```

Metoda sprawdza poprawnosc algorytmu.

Wczytywane sa poprawne dane wynikowe, a nastepnie sa one porownywane z tymi otrzymanymi przez wykonanie algorytmu

Returns

0 - gdy algorytm jest poprawny, -1 - gdy nie.

3.1.2.2 void Dzialanie::wczytajDaneWejsciowe (string nazwa, stos_tab stos)

metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.

Format danych w pliku jest nastepujacy: pierwszy wiersz - ilosc elementow, a nastepnie w kolumnie kolejne wartosci tablicy.

Parameters

nazwaPliku - nazwa pliku do otwarcia

Returns

void

3.1.2.3 LARGE_INTEGER Dzialanie::wlaczStoper ()

Metoda uruchamia pomiar czasu.

Czas jest mierzony w milisekundach

Returns

czas, w ktorym zostal wlaczony stoper

3.1.2.4 void Dzialanie::wykonajAlgorytm ()

Metoda wykonuje algorytm na danych wejsciowych (tablicy)

Algorytm do wykonania : pomnoz kazdy element razy 2.

Returns

void

3.1.2.5 LARGE_INTEGER Dzialanie::wylaczStoper ()

Metoda konczy pomiar czasu.

Czas jest mierzony w milisekundach

Returns

czas, w ktorym stoper zostal wylaczony

The documentation for this class was generated from the following files:

- · inc/dzialanie.hh
- · src/dzialanie.cpp

3.2 kolejka_list Class Reference

Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste.

```
#include <kolejka_list.hh>
```

Public Member Functions

· void enqueue (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

void dequeue (int *a)

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

· bool isempty ()

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

• int size ()

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

void wyswietl ()

3.2.1 Detailed Description

Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste.

3.2.2 Member Function Documentation

3.2.2.1 void kolejka_list::dequeue (int *a)

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

Parameters

```
a - wskaznik do zmiennej, do ktorej sciagamy wartosc.
```

3.2.2.2 void kolejka_list::enqueue (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

Parameters

```
element - element, ktory zostanie dodany do stosu
```

3.2.2.3 bool kolejka_list::isempty ()

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

Returns

lista.empty();

```
3.2.2.4 int kolejka_list::size ( )
```

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

Returns

rozmiar stosu.

The documentation for this class was generated from the following files:

- · inc/kolejka_list.hh
- · src/kolejka_list.cpp

3.3 kolejka_tab Class Reference

Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste.

```
#include <kolejka_tab.hh>
```

Public Member Functions

- kolejka_tab (int zwieksz)
- void enqueue (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

void dequeue (int *a)

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

• bool isempty ()

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

• int size ()

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

void wyswietl ()

3.3.1 Detailed Description

Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste.

3.3.2 Member Function Documentation

```
3.3.2.1 void kolejka_tab::dequeue ( int * a )
```

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

Parameters

```
a - wskaznik do zmiennej, do ktorej sciagamy wartosc.
```

3.3.2.2 void kolejka_tab::enqueue (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

Parameters

```
element - element, ktory zostanie dodany do stosu
```

```
3.3.2.3 bool kolejka_tab::isempty ( )
```

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

Returns

```
lista.empty();
```

```
3.3.2.4 int kolejka_tab::size ( )
```

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

Returns

rozmiar stosu.

The documentation for this class was generated from the following files:

- · inc/kolejka_tab.hh
- · src/kolejka_tab.cpp

3.4 Kontener Class Reference

Klasa Dane.

```
#include <kontener.hh>
```

Public Member Functions

• unsigned int wez_rozmiar ()

Metoda zwracajaca rozmiar tablicy.

vector< int > & wez_dane ()

Metoda zwracajaca referencje do tablicy danych Metoda pozwala na dostep do tablicy i jej modyfikacje.

void wczytajDane (string nazwaPliku)

metoda wczytuje dane do tablicy z pliku

void zamien_elementy (unsigned int i, unsigned int j)

Metoda zamienia ze soba dwa elementy tablicy.

void odwroc_kolejnosc ()

Metoda odwraca zawartosc tablicy.

• void dodaj_element (int e)

Metoda dodaje element na koniec tablicy.

• void dodaj_elementy (Kontener tab)

Metoda dodaje na koniec tablicy zawartosc innej tablicy.

int & operator[] (int index)

Operator indeksujacy tablice.

Kontener & operator+ (Kontener tab)

Operator dodawania tablic Operator pozwala na dodanie 2 tablic.

Kontener & operator= (Kontener tab)

Operator przypisania Operator pozwala na przypisanie do tablicy zawartosci innej tablicy.

bool operator== (Kontener tab)

Operator porownania 2 tablic Operator pozwala na porownanie 2 tablic. Sprawdza on czy sa takie same pod wzgledem zawartosci.

Friends

ostream & operator<< (ostream &out, Kontener Tab)

Operator wwypisywania Metoda pozwala na wypisanie zawartosci tablicy na standardowe wyjscie.

3.4.1 Detailed Description

Klasa Dane.

Klasa posiada 2 pola: -tablice (vector), -rozmiar tablicy.

3.4.2 Member Function Documentation

3.4.2.1 void Kontener::dodaj_element (int e)

Metoda dodaje element na koniec tablicy.

Parameters

e - wartosc elementu

3.4.2.2 void Kontener::dodaj_elementy (Kontener tab)

Metoda dodaje na koniec tablicy zawartosc innej tablicy.

Parameters

tablica,ktora bedzie dodana na koniec

3.4.2.3 Kontener & Kontener::operator+ (Kontener tab)

Operator dodawania tablic Operator pozwala na dodanie 2 tablic.

```
\param tablica do dodania
```

Returns

dwie polaczone tablice

3.4.2.4 Kontener & Kontener::operator= (Kontener tab)

Operator przypisania Operator pozwala na przypisanie do tablicy zawartosci innej tablicy.

Parameters

tablica,ktora	przypisujemy	
---------------	--------------	--

3.4.2.5 bool Kontener::operator== (Kontener tab)

Operator porownania 2 tablic Operator pozwala na porownanie 2 tablic. Sprawdza on czy sa takie same pod wzgledem zawartosci.

Parameters

```
tablica,z ktora bedziemy porownywac
```

Returns

```
true - tablice sa identyczne false - tablice sa rozne
```

3.4.2.6 int & Kontener::operator[] (int index)

Operator indeksujacy tablice.

Parameters

index	- indeks, ktorego referencja zostanie zwrocona

Returns

referencja do zadanego indeksu

3.4.2.7 void Kontener::wczytajDane (string nazwaPliku)

metoda wczytuje dane do tablicy z pliku

Format danych w pliku jest nastepujacy: pierwszy wiersz - ilosc elementow, a nastepnie w kolumnie kolejne wartosci tablicy.

Parameters

```
nazwaPliku | - nazwa pliku do otwarcia
```

Returns

void

3.4.2.8 vector<int>& Kontener::wez_dane() [inline]

Metoda zwracajaca referencje do tablicy danych Metoda pozwala na dostep do tablicy i jej modyfikacje.

Returns

referencja do tablicy

3.4.2.9 unsigned int Kontener::wez_rozmiar () [inline]

Metoda zwracajaca rozmiar tablicy.

Returns

rozmiar tablicy

3.4.2.10 void Kontener::zamien_elementy (unsigned int i, unsigned int j)

Metoda zamienia ze soba dwa elementy tablicy.

Parameters

```
i,j - indeksy, ktore zostana zamienione
```

The documentation for this class was generated from the following files:

- · inc/kontener.hh
- · src/kontener.cpp

3.5 stos list Class Reference

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste.

```
#include <stos_list.hh>
```

Public Member Functions

void push (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

void pop (int *a)

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

• bool isempty ()

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

• int size ()

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

void wyswietl ()

3.5.1 Detailed Description

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste.

3.5.2 Member Function Documentation

```
3.5.2.1 bool stos_list::isempty ( )
```

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

Returns

lista.empty();

```
3.5.2.2 void stos_list::pop ( int *a )
```

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

Parameters

```
a - wskaznik do zmiennej, do ktorej sciagamy wartosc.
```

3.5.2.3 void stos_list::push (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

Parameters

```
element | - element, ktory zostanie dodany do stosu
```

```
3.5.2.4 int stos_list::size ( )
```

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

Returns

rozmiar stosu.

The documentation for this class was generated from the following files:

- · inc/stos_list.hh
- · src/stos_list.cpp

3.6 stos_tab Class Reference

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice.

```
#include <stos_tab.hh>
```

Public Member Functions

- stos tab (int zwieksz)
- · void push (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec tablicy. W przypadku gdy tablica jest pelna zostaje ona powiekszona o 1 element.

void pop (int *a)

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu tablicy. Z kazdorazowym pobraniem danych rozmiar tablicy jest zmniejszany o 1.

· bool isempty ()

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty. Sprawdzenie polega na odczytaniu zmiennej rozmiar,

ktora pamieta ile jest elementow na stosie.

• int size ()

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar stosu. Rozmiar stosu przechowywany jest w zmiennej rozmiar.

· void wyswietl ()

3.6.1 Detailed Description

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice.

3.6.2 Member Function Documentation

```
3.6.2.1 bool stos_tab::isempty ( )
```

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty. Sprawdzenie polega na odczytaniu zmiennej rozmiar,

ktora pamieta ile jest elementow na stosie.

Returns

```
true - rozmiar == 0 false - rozmiar > 0.
```

```
3.6.2.2 void stos_tab::pop ( int *a )
```

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu tablicy. Z kazdorazowym pobraniem danych rozmiar tablicy jest zmniejszany o 1.

Parameters

```
a - wskaznik do zmiennej, do ktorej sciagamy wartosc.
```

3.6.2.3 void stos_tab::push (int element)

Metoda dodajaca element na stos Metoda dodaje element na koniec tablicy. W przypadku gdy tablica jest pelna zostaje ona powiekszona o 1 element.

Parameters

```
element | - element, ktory zostanie dodany do stosu
```

```
3.6.2.4 int stos_tab::size ( )
```

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar stosu. Rozmiar stosu przechowywany jest w zmiennej rozmiar.

Returns

rozmiar stosu.

The documentation for this class was generated from the following files:

- · inc/stos tab.hh
- · src/stos_tab.cpp

3.7 Tester Class Reference

Klasa Tester Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokładniejszym pomiarem.

3.7 Tester Class Reference 15

#include <tester.hh>

Public Member Functions

• Tester ()

konstruktor klasy Konstruktor inicjuje wartosci: -powtorzenia (ile razy wykonywac algorytm) -ilosc (ile mamy zestawow dancyh) -wejscie (nazwa pliku z danymi wejsciowymi) -wynik (nazwa pliku z poprawnym wynikiem algorytmu)

void otworzPlik ()

Metoda otwierajaca referencje do pliku CSV.

void zamknijPlik ()

Metoda otwierajaca referencje do pliku CSV.

· void symulacja ()

Metoda symulujaca badanie algorytmu Metoda wykonuje symulacje dzialania algorytmu. Wykonuje algorytm zadana liczbe razy dla zadanych zestawow danych. Wyniki zapisuje do pliku CSV o nazwie rezultat.csv Format zapisu: rozmiar_problemu,ilosc_powtorzen,sredni_czas.

void zamienNazwy (int numer)

Metoda aktualizuje nazwe plikow wejsciowych Format plikow wejsciowych jest scisle okreslony. Dane wejsciowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.

3.7.1 Detailed Description

Klasa Tester Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokładniejszym pomiarem.

3.7.2 Member Function Documentation

3.7.2.1 void Tester::zamienNazwy (int numer)

Metoda aktualizuje nazwe plikow wejsciowych Format plikow wejsciowych jest scisle okreslony. Dane wejsciowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.

Parameters

numer	- % zostaje zamieniony na wartosc 'numer'

The documentation for this class was generated from the following files:

- · inc/tester.hh
- src/tester.cpp

Index

operator==

dequeue	Kontener, 10
kolejka list, 7	
kolejka_tab, 8	рор
dodaj_element	stos_list, 12
Kontener, 10	stos_tab, 14
dodaj elementy	push
Kontener, 10	stos_list, 13
Dzialanie, 5	stos_tab, 14
sprawdz, 5	- ·
wczytajDaneWejsciowe, 6	size
wlaczStoper, 6	kolejka_list, 7
wykonajAlgorytm, 6	kolejka_tab, 9
wykonajaigoryim, o wylaczStoper, 6	stos_list, 13
wyłaczotoper, o	stos_tab, 14
engueue	sprawdz
enqueue kolejka_list, 7	Dzialanie, 5
kolejka_tab, 8	stos list, 12
KOIEJKa_IaD, o	isempty, 12
isomoty	pop, 12
isempty	push, 13
kolejka_list, 7	size, 13
kolejka_tab, 9	stos_tab, 13
stos_list, 12	isempty, 14
stos_tab, 14	pop, 14
kolojka list 7	push, 14
kolejka_list, 7	size, 14
dequeue, 7	3123, 11
enqueue, 7	Tester, 14
isempty, 7	zamienNazwy, 15
size, 7	, ,
kolejka_tab, 8	wczytajDane
dequeue, 8	Kontener, 11
enqueue, 8	wczytajDaneWejsciowe
isempty, 9	Dzialanie, 6
size, 9	wez_dane
Kontener, 9	Kontener, 11
dodaj_element, 10	wez_rozmiar
dodaj_elementy, 10	Kontener, 11
operator+, 10	wlaczStoper
operator=, 10	Dzialanie, 6
operator==, 10	wykonajAlgorytm
wczytajDane, 11	Dzialanie, 6
wez_dane, 11	wylaczStoper
wez_rozmiar, 11	Dzialanie, 6
zamien_elementy, 12	Dziaianie, V
	zamien elementy
operator+	Kontener, 12
Kontener, 10	zamienNazwy
operator=	Tester, 15
Kontener, 10	