Piotr KaroÄšÂ, 241 626 WrocÄšÂaw, dn. 03.04.2019

Zadanie projektowe 2.

Badanie efektywnoÄšÂci algorytmÄĹw grafowych w zaleÄšĹşnoÄšÂci od rozmiaru instancji oraz sposobu reprezentacji grafu w pamiĂÂci komputera.

PROWADZĂÂCY:
dr JarosÄšÂaw Mierzwa

Spis treści

1 ZaÄšÂoÄšĹşenia projektowe

1.1 Cel

Celem projektu jest zbadanie efektywnoÄšÂci algorytmÄĹw Kruskala, Prima, Dijkstry i Bellmana-Forda w zaleÄšĹşnoÄšÂci od sposobu reprezentacji grafu i wielkoÄšÂci instancji.

1.2 Technologie

Do implementacji wymienionych struktur uÄšĹşyto jÄÂzyka ***Kotlin*** w wersji ***Native***, ktÄĹra jest kompilowana do kodu maszynowego danej platformy.

1.3 Przebieg eksperymentu

Badania przeprowadzone zostaÄšÂy dla wierzchoÄšÂkÄĹw w liczbie: 10, 100, 1000, 10000, 30000, dla kaÄšĹşdej liczby w gÄÂstoÄšÂciach 25%, 50%, 75%, 99% osobno dla reprezentacji macierzowej i listowej. KaÄšĹşdy test wykonano 50 razy, a czas uÄšÂredniono z wszystkich prÄĹb.

2 KrÄĹtki opis algorytmÄĹw

2.1 Algorytm Kruskala

Czas dziaÄšÂania algorytmu Kruskala dla grafu $G = (V, E)$ zaleÄšĹşy od sposobu implementacji zbiorÄĹw rozÄšÂÄÂcznych. W tym projekcie wykorzystano implementacjÄÂ lasu zbiorÄĹw rozÄšÂÄÂcznych z ÄšÂÄÂczeniem wedÄšÂug rangi i z kompresjÄÂ ÄšÂcieÄšĹşek.

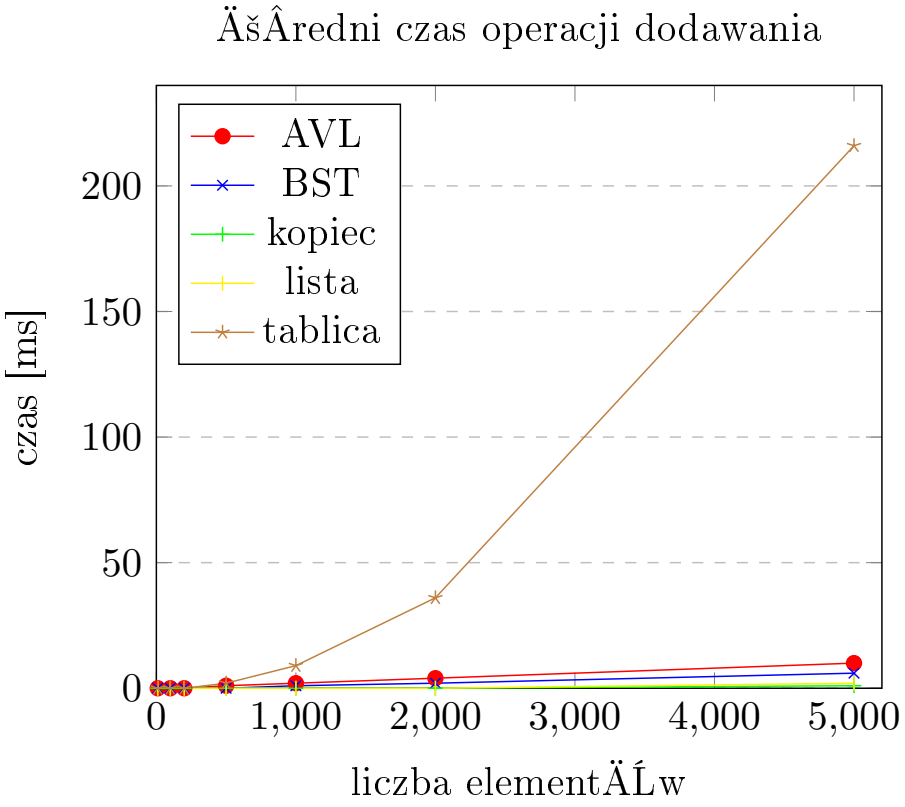
CaÄšÂkowity czas dziaÄšÂania algorytmu wynosi $O(Elog_2E)$

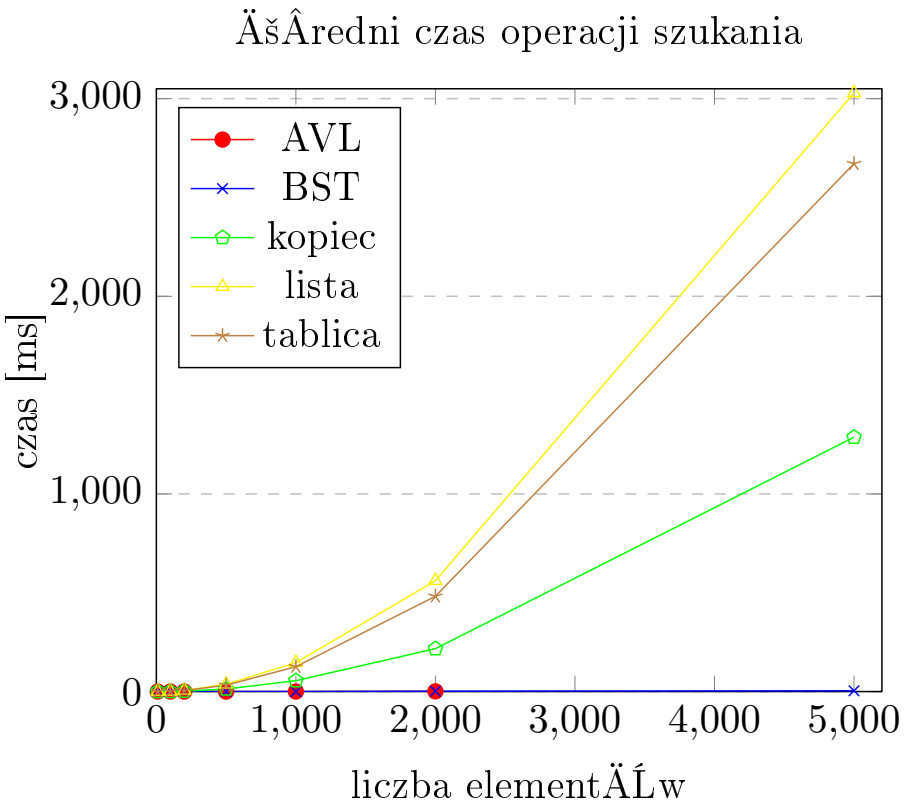
2.2 Lista

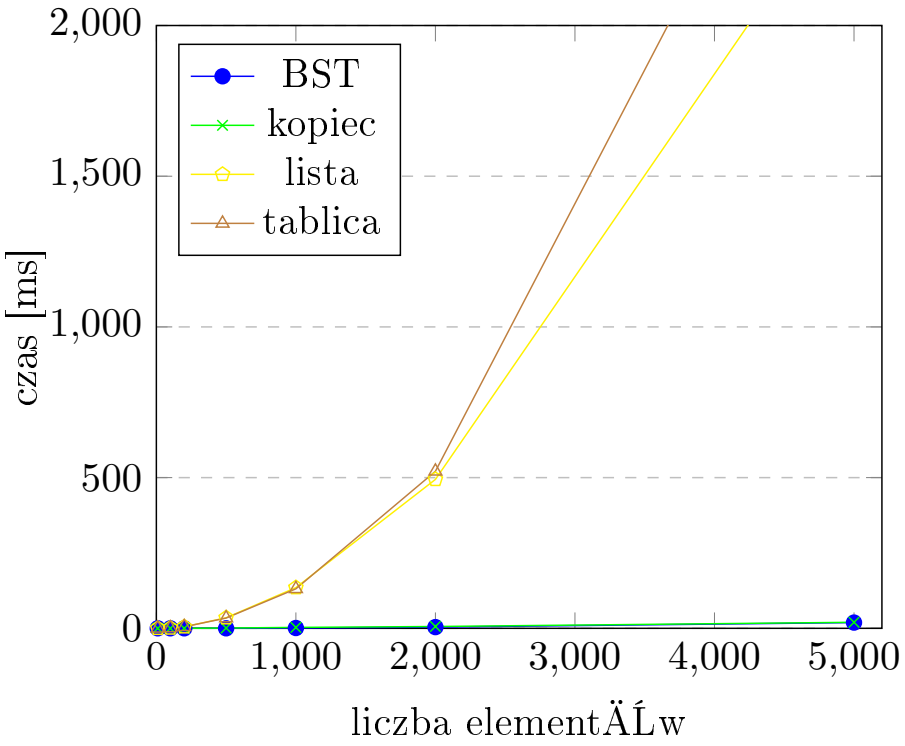
2.3 Kopiec binarny (maksymalny)

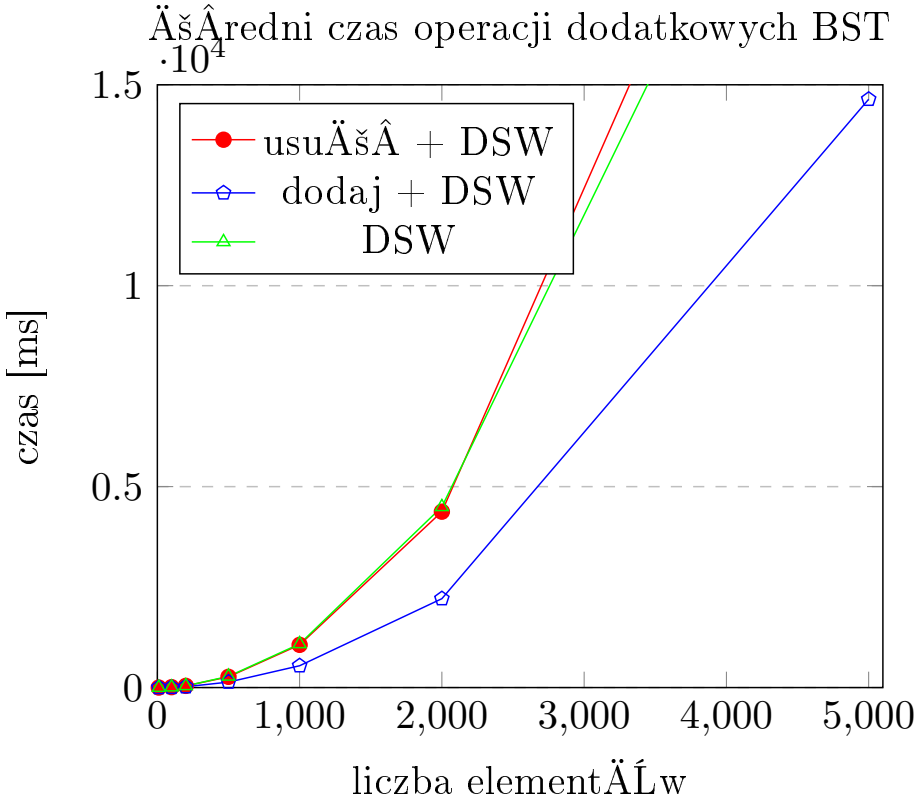
3 Wyniki

3.1 Wykresy




ÄšÂredni czas operacji szukania
3,000
2,000
1,000
0
czas [ms]
AVL
BST
kopiec
lista
tablica
0
1,000
2,000
3,000
4,000
5,000
liczba elementÄĹw


ÄšÂredni czas operacji usuwania
2,000
1,500
1,000
500
0
czas [ms]
BST
kopiec
lista
tablica
0
1,000
2,000
3,000
4,000
5,000
liczba elementÄĹw



3.2 Tabele

Tablica 1: Wyniki operacji dodawania w milisekundach

*	Tablica	Lista	Kopiec	Bst	Avl
10	0	0	0	0	0
100	0	0	0	0	0
200	0	0	0	0	0
500	2	0	0	0	1
1000	9	0	0	1	2
2000	36	0	0	2	4
5000	216	2	1	6	10

Tablica 2: Wyniki operacji szukania w milisekundach

*	Tablica	Lista	Kopiec	Bst	Avl
10	0	0	0	0	0
100	1	1	0	0	0
200	5	6	0	0	0
500	32	37	0	0	0
1000	126	146	0	1	0
2000	482	561	0	2	1
5000	2671	3029	1	6	2

Tablica 3: Wyniki operacji usuwania w milisekundach

*	Tablica	Lista	Kopiec	Bst	Avl
10	0	0	0	0	*
100	1	1	0	0	*
200	5	5	0	0	*
500	34	35	1	0	*
1000	131	135	3	1	*
2000	521	494	6	4	*
5000	3185	2512	20	19	*

Tablica 4: Wyniki operacji specjalnych BST

	RÄĹwnowaÄšĹşenie	Usuwanie + DSW	Dodawanie + DSW
10	0	0	0
100	10	10	5
200	42	42	21
500	279	267	136
1000	1086	1063	543
2000	4497	4377	2212
5000	28523	26293	15005

4 Podsumowanie

Zaimplementowane algorytmy nie sĂÂ optymalne. W wiĂÂkszoÄšÂci jednak zaÄšÂoÄšĹşona zÄšÂoÄšĹşonoÄšÂĂÂ obliczeniowa sprawdziÄšÂa siĂÂ. Najbardziej obciĂÂÄšĹşa-

jĂÂce byÄšÂy operacje na drzewie BST z wykorzystaniem algorytmu DSW. RÄĹwnowaÄšĹşenie drzewa po kaÄšĹşdym wstawieniu wĂÂzÄšÂa nie jest dobrym pomysÄšÂem. MoÄšĹşna to robiĂÂ co kilka, kilkanaÄšÂcie wstawieÄšÂ – takie niezrÄĹwnowaÄšĹşenie nie wpÄšÂynie bardziej na zÄšÂoÄšĹşonoÄšÂĂÂ w przypadku innych operacji.

Literatura

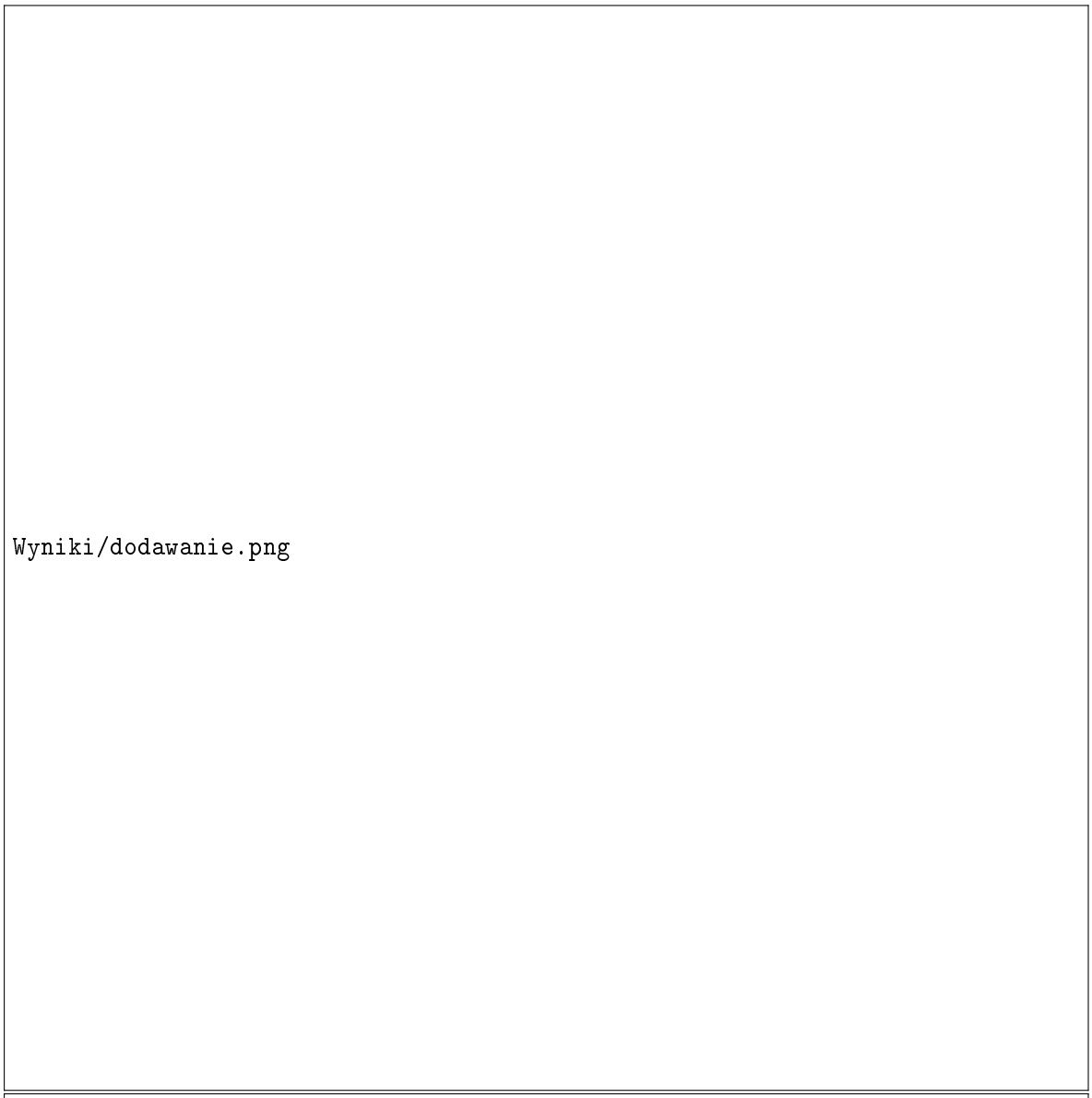
[1] T. Cormen, C. Leiserson, R. Rivest *Wprowadzenie do algorytmÄĹw*, Wydawnictwa Naukowo-Techniczne Warszawa, Wyd. IV, 2004

[2] [1] str. 304.

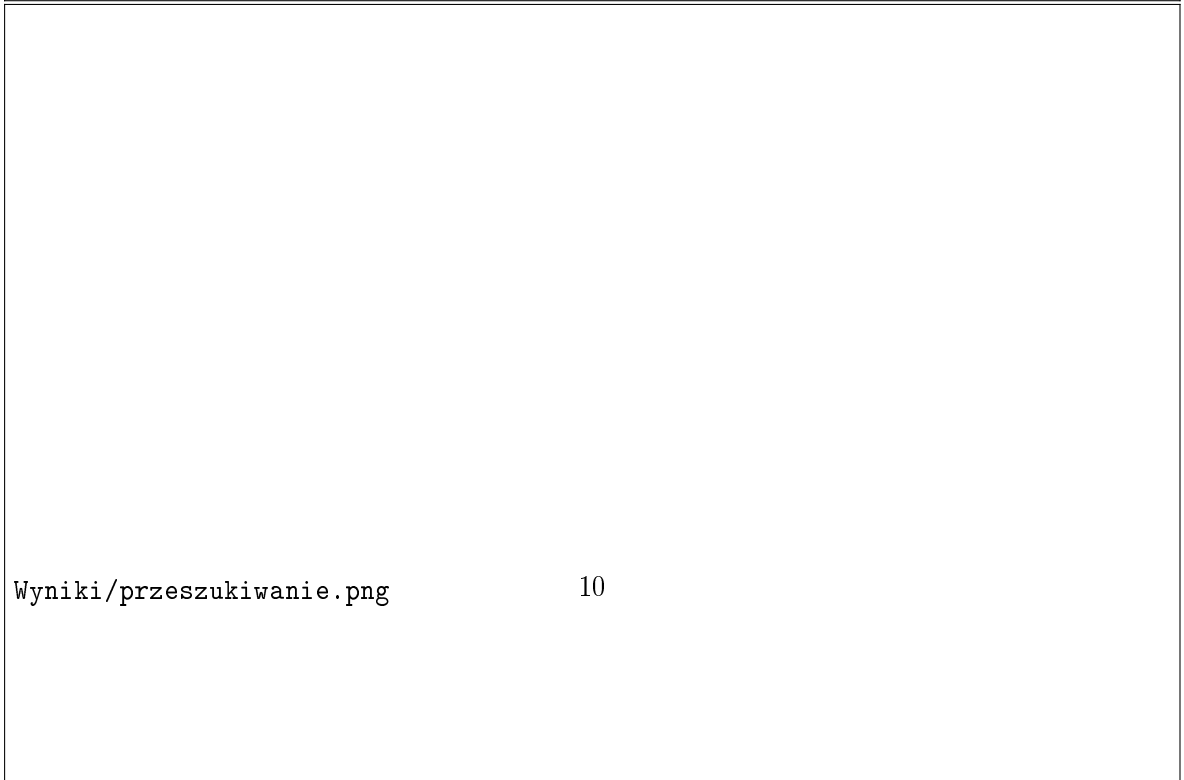
[3] `tomasz.kaplon.staff.iiar.pwr.wroc.pl/`, strona dr Tomasza KapÄšÂona

[4] `kotlinlang.org/docs/reference/native-overview.html`, dokumentacja jÄÂzyka Kotlin/Native

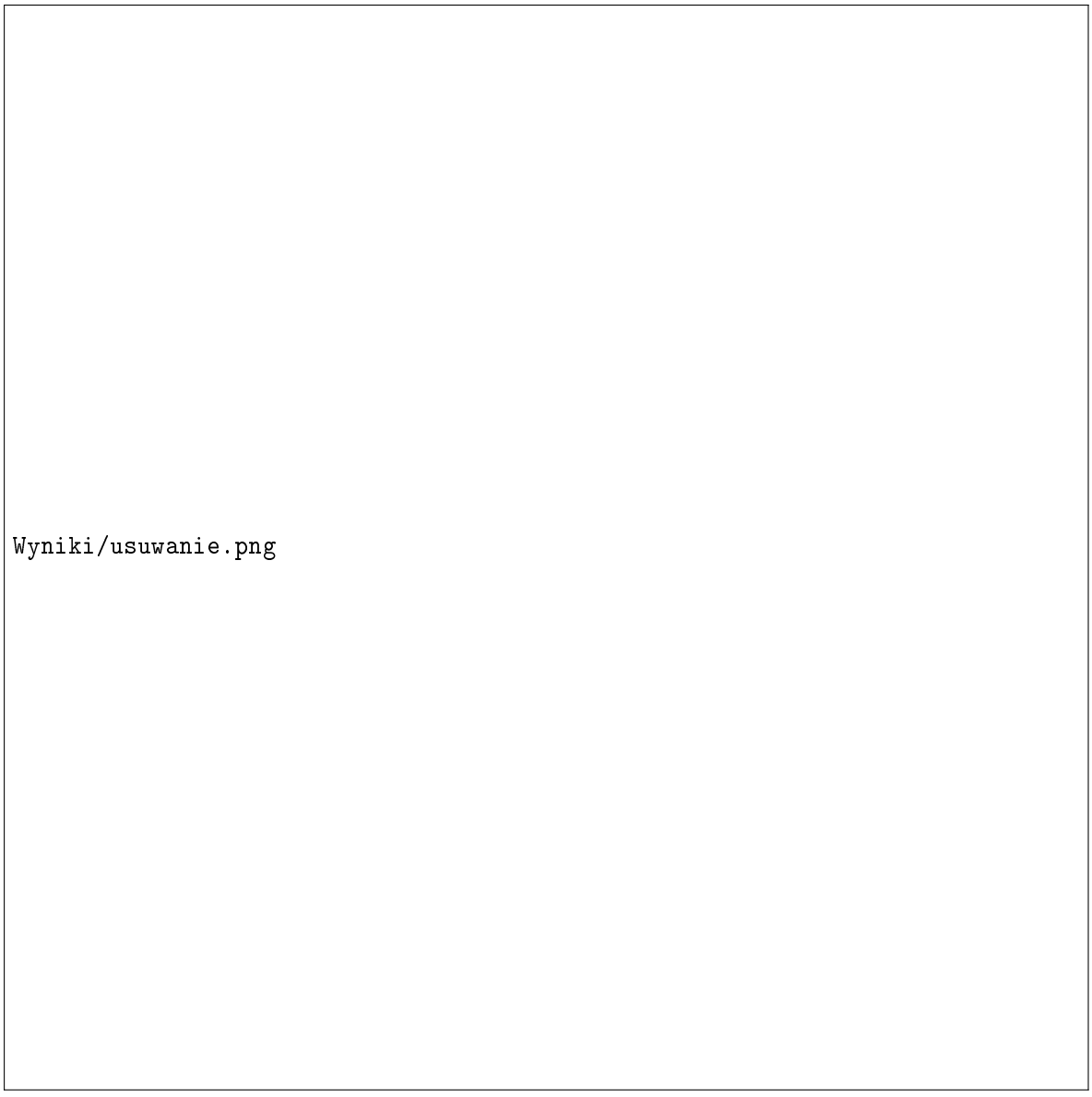
[5] `eduinf.waw.pl`, materiaÄšÂy na stronie I LO w Tarnowie



Wyniki/dodawanie.png



Wyniki/przeszukiwanie.png


Wyniki/usuwanie.png