Projektowanie algorytmów i metody sztucznej inteligencji

Dawid Marszałkiewicz 218665

10 kwietnia 2016

1 Zadanie

- Umieść w wybranej strukturze, zadane rozmiary problemów N= $10^1, 10^2, 10^3, 10^6, 10^8$.
- Zaimplementuje algorytm Quick sort.
- Zmierz czas sortowania dla tablic o rozmiarze N (wartość pobrana ze struktury).

2 Pomiary

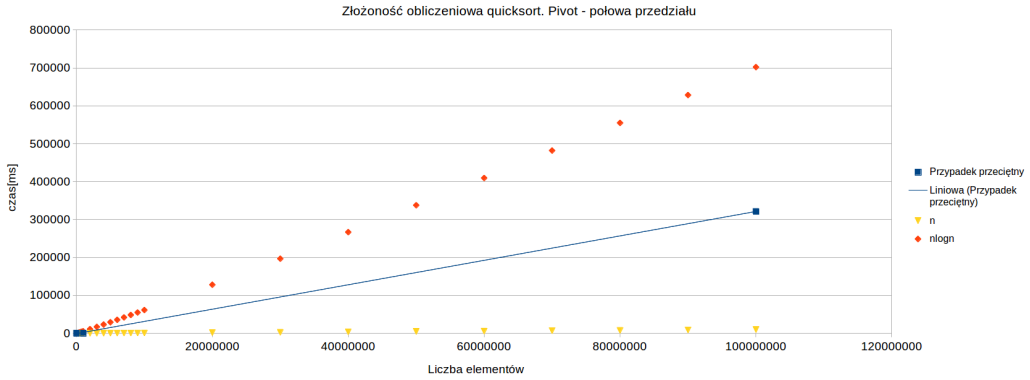
2.1 Przypadek przeciętny

Tabela 1: Pomiary czasów dla przypadku przeciętnego

l. elementów	Strategia 1	Strategia 2	Strategia 3
	[ms]	[ms]	[ms]
10	0,0025	0,0030	0,0056
100	0,0244	0,0315	0,0689
1000	0,3370	0,4324	0,2946
1000000	503,6726	490,3309	510,1875
100000000	321496,4835	314639,9734	313564,9930

2.1.1 Strategia pierwsza

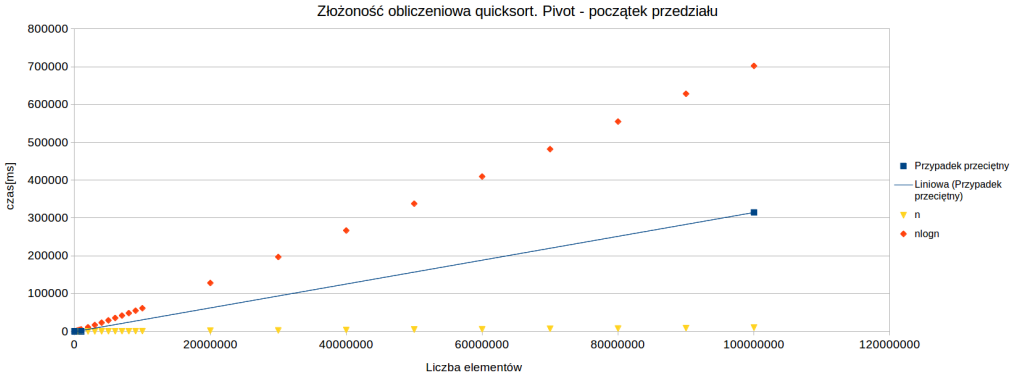
Strategia ta polega na wybieraniu pivota jako element w połowie przedziału.



Rysunek 1: Wykres złożoności obliczeniowej

2.1.2 Strategia druga

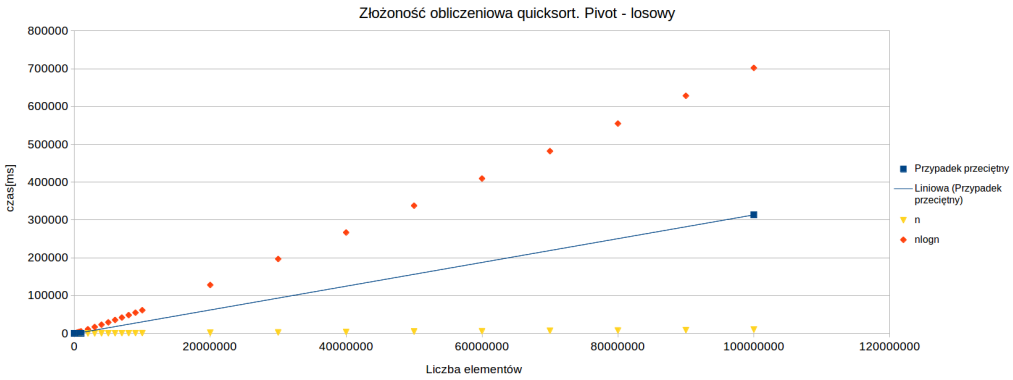
Strategia ta polega na wybieraniu pivota jako element na początku przedziału.



Rysunek 2: Wykres złożoności obliczeniowej

2.1.3 Strategia trzecia

Strategia ta polega na wybieraniu pivota jako losowy element z przedziału.



Rysunek 3: Wykres złożoności obliczeniowej

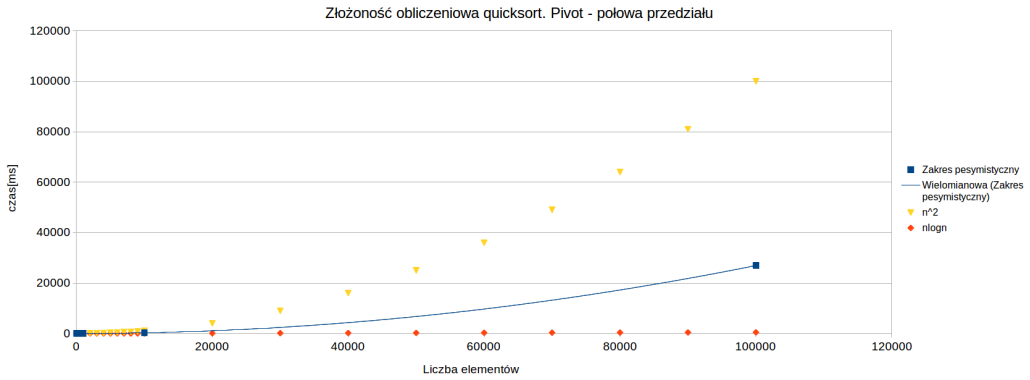
2.2 Przypadek pesymisityczny

Tabela 2: Pomiary czasów dla przypadku pesymistycznego

l. elementów	Strategia 1	Strategia 2	Strategia 3
	[ms]	[ms]	[ms]
10	0,0030	0,0048	0,0032
100	0,0581	0,1158	0,0368
1000	3,5608	3,4189	3,0519
10000	267,7446	268,1385	269,6280
100000	26947,5106	27571,1318	26607,2129

2.2.1 Strategia pierwsza

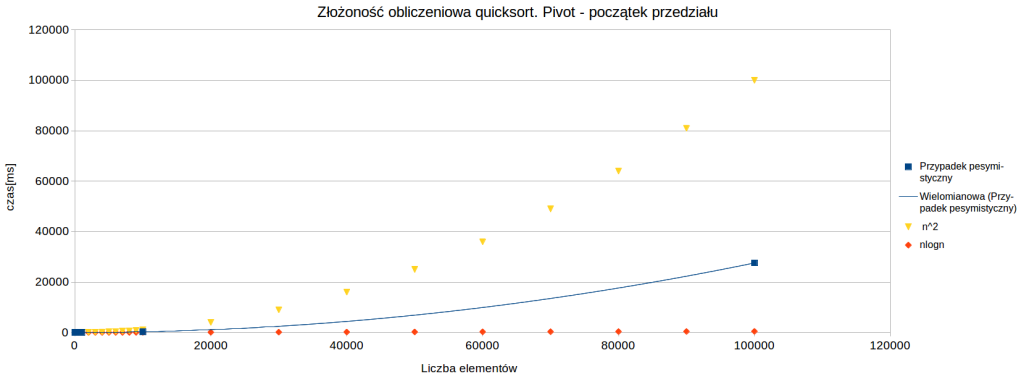
Dla tej strategii, przypadek pesymistyczny występował dla tablicy, w której elementy są sobie równe.



Rysunek 4: Wykres złożoności obliczeniowej

2.2.2 Strategia druga

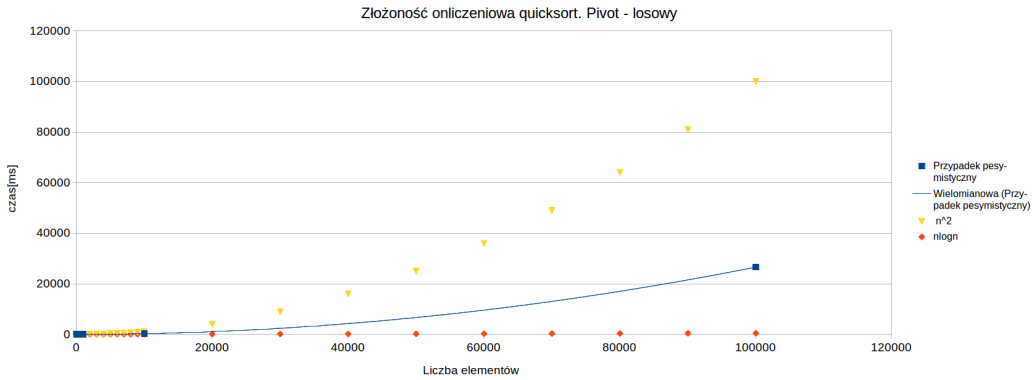
Dla tej strategii, przypadek pesymistyczny występował dla tablicy już posortowanej.



Rysunek 5: Wykres złożoności obliczeniowej

2.2.3 Strategia trzecia

Dla tej strategii, przypadek pesymistyczny występował dla tablicy, w której elementy są sobie równe.



Rysunek 6: Wykres złożoności obliczeniowej

3 Wnioski

- Algorytm *Quick sort* w przypadku przeciętnym dla wszystkich trzech strategii wyboru pivota ma złożoność obliczeniową O($nlogn$).

- Algorytm *Quick sort* w przypadku pesymistycznym dla wszystkich trzech strategii wyboru pivota ma złożoność obliczeniową O(n^2).

- Wybieranie pivota jako połowa przedziału oraz losowy wybór pivota są najbardziej odporne na przypadek pesymistyczny. W przypadku posortowanej tablicy złożoność obliczeniową O($nlogn$). Najmniejszą odporność na przypadek pesymistyczny uzyskujemy podczas wybierania pivota jako element skrajny. Dla posortowanej tablicy *Quick sort* upodobnia się do *Bubble sort*, a jego złożoność obliczeniowa wynosi O(n^2)

- Liczby do tablicy były wybierane losowo z przedziału 0...MAX_RAND, czas sortowania malał wraz ze wzrostem szerokości tego przedziału. Wynika to z tego, że dla mniejszych przedziałów jest większe prawdopodobieństwo wystąpienia przypadku pesymistycznego na poziomie lokalnym (np. część tablicy jest już posortowana lub są to elementy o takiej samej wartości).

- Wybór pivota ma wpływ na wystąpienie przypadku pesymistycznego. Największe prawdopodobieństwo wystąpienia tego przypadku ma strategia druga, natomiast ze względu na losowość pivota najmniejsze prawdopodobieństwo posiada strategia trzecia.

- Jako strukturę, na którą wpisuję zadane rozmiary tablicy wybrałem listę ze względu na większa funkcjonalność niż kolejka i stos, co ułatwiało pracę na etapie testowania oprogramowania. Lista jak i pozostałe strukture były dobrze zadokumentowane oraz posiadały metody zgodne z ogólnie przyjętą zasadą.

- Algorytm *Merge sort* wykonywał się ze złożonością obliczeniową O($nlogn$), jednak wymaga większej złożoności pamięciowej, niż *Quick sort*. Algorytm ten lepiej sobie radzi z przypadkami (złożoność obliczeniowa O($nlogn$)), w których *Quick sort* ma złożoność obliczeniową O(n^2).