 Politechnika Wrocławska	Projektowanie algorytmów i metody sztucznej inteligencji	Data ćwiczenia: 27.04.2017 Data sprawozdania: 17.05.2017
	Badanie czasu wyszukiwania elementów w tablicy asocjacyjnej	Rafał Borysionek, 226262 Automatyka i Robotyka Wydział Elektroniki
Prowadzący: Mgr inż. Andrzej Wytyczak-Partyka		Grupa: E02-18o

1. Cel ćwiczenia

Celem ćwiczenia jest zbadanie złożoności obliczeniowej wyszukiwania elementów w tablicy asocjacyjnej pod kątem używanego algorytmu haszującego.

2. Sposób badania

Zaimplementowano dwie funkcje haszowania dla własnego typu danych:

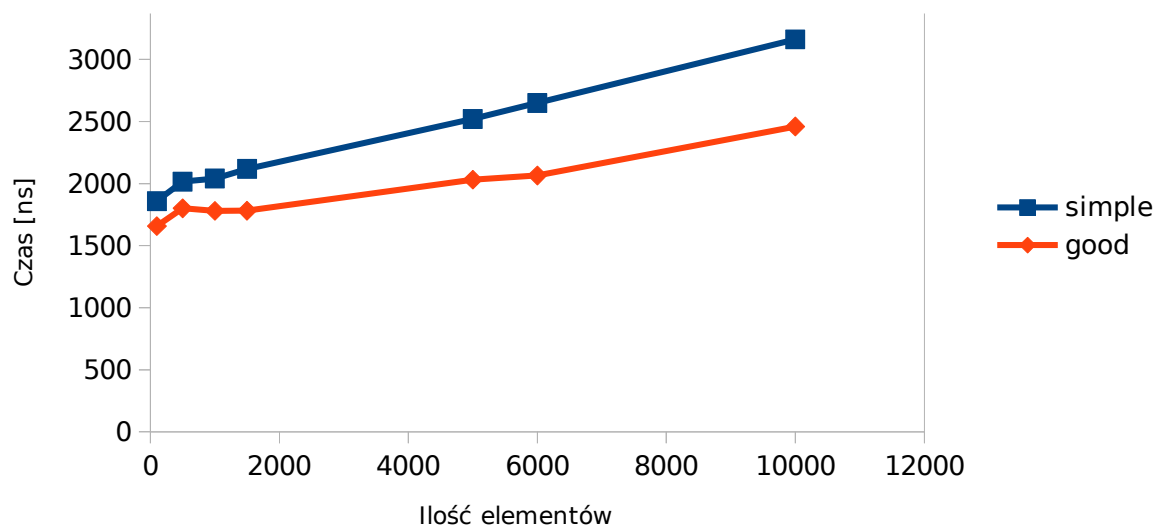
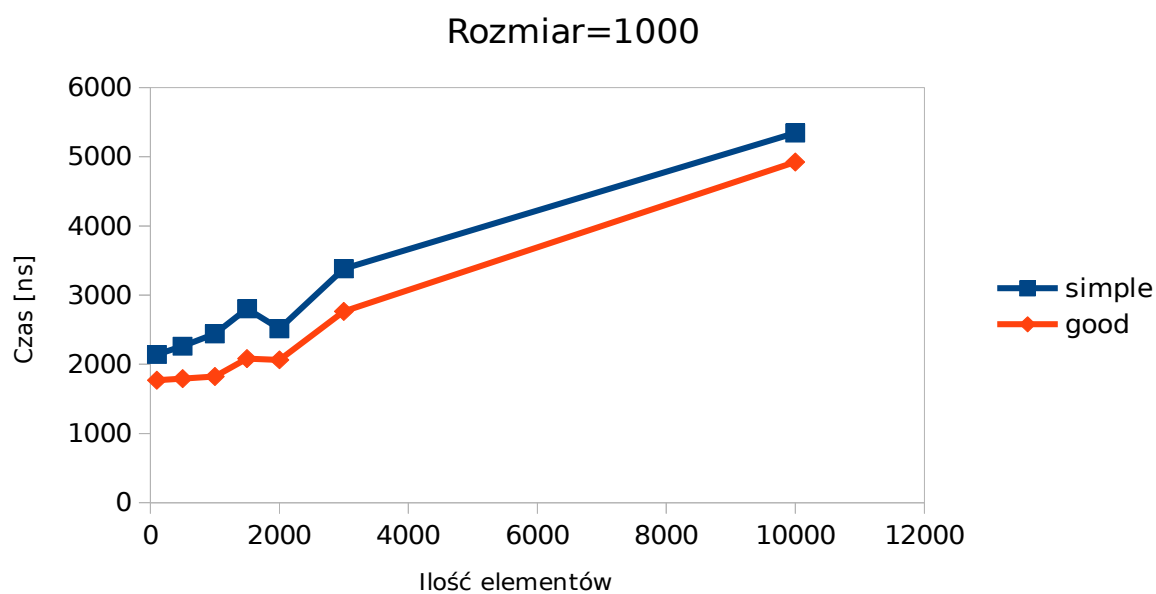
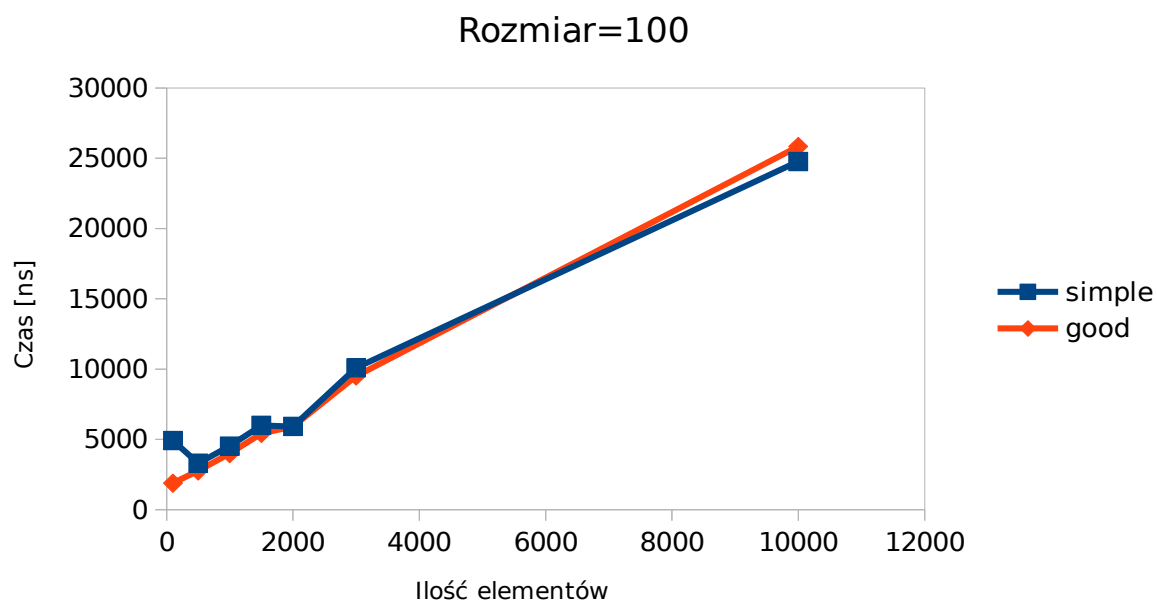
-Haszowanie modularne: $h(klucz) = klucz \times modulo(rozmiar)$

-Haszowanie przez mnożenie: $h(klucz) = podłoga(rozmiar \times A \times modulo(1))$

gdzie: $A = \frac{\sqrt{5} - 1}{2}$ (optymalna stała, dla której algorytm daje lepsze rezultaty)

Dane algorytmy badano dla wyszukania konkretnego elementu, który był umieszczany na końcu listy (przy ewentualnych kolizjach).

3. Wykresy



4. Wnioski

1. Tablica asocjacyjna powinna być używana tam, gdzie raczej nie przewiduje się powiększania danych. Oprócz tego, użytkownik powinien wiedzieć już na początku ile danych chce zawrzeć w tablicy żeby wybrać optymalne rozwiązanie pomiędzy długim czasem odczytu a dużym rozmiarem samej tablicy asocjacyjnej.
2. Optymistyczna złożoność obliczeniowa obydwu badanych algorytmów wynosi $O(1)$ czyli jest stała. Pesymistyczny przypadek złożoności to $O(n)$ czyli jest liniowa. Spowodowane jest to występowaniem kolizji (dla dwóch par przypisane są te same indeksy za pomocą funkcji haszującej).
3. Złożoność obliczeniowa nie zależy od ilości elementów tylko od stopnia zapełnienia tablicy asocjacyjnej w kontekście jej rozmiaru. Im większy stopień zapełnienia (tak jak na wykresie 1) tym złożoność obliczeniowa szybciej zaczyna stawać się funkcją liniową.
4. Haszowanie modularne jest mniej wydajne od haszowania przez mnożenie. Ukazują to wykresy 2 i 3. W lepszym haszowaniu, złożoność stała $O(1)$ występuje dłużej podczas gdy w haszowaniu modularnym, algorytm przybiera złożoność $O(n)$ przy mniejszym stopniu zapełnienia tablicy, zatem jest gorszy.
5. Haszowanie przez mnożenie nie jest jednak idealną funkcją haszującą. Aby funkcja była idealna, nie może występować problemu kolizji. Aby to spełnić, funkcja, musi spełniać warunek równomiernego haszowania. Polega on na tym, że losowo wybrany klucz jest z jednakowym prawdopodobieństwem odwzorowany na każdą z m pozycji. Ten warunek musi być spełniony niezależnie od tego gdzie zostaną odwzorowane inne klucze. Ponadto, powinna ona przypisywać „bliskim” wartościom kluczy- oddalone indeksy. Dobór funkcji haszującej powinien się opierać na danym uniwersum potencjalnie używanych kluczy, i na jego podstawie należy dobierać funkcję haszującą.