

ADT - kolejki i stosy, testowanie algorytmów dodawania

Marcin Ochman

15 marca 2014

1 Opis problemu

Kolejki i stosy to jedne z najczęściej używanych abstrakcyjnych typów danych. Można je zaimplementować dwojako:

- za pomocą listy
- za pomocą tablicy

Tutaj pojawia się problem wyboru odpowiedniej implementacji. Główne pytanie przy jej wyborze, które powinniśmy sobie zadać brzmi „na czym najbardziej nam zależy”. Lista ma tę przewagę, że operacje dodawania na koniec i początek mają złożoność $\mathcal{O}(1)$. Wadą jest wymagana dodatkowa pamięć przeznaczona na wskaźniki na następny element. Z tablicą jest nieco inaczej. Złożoność obliczeniowa dodawania i usuwania z początku lub końca tablicy zależy od zastosowanej strategii alokacji pamięci. Tyczy się to również wymagań pamięciowych. Najbardziej znanymi są:

- strategia podwajania
- strategia inkrementacji

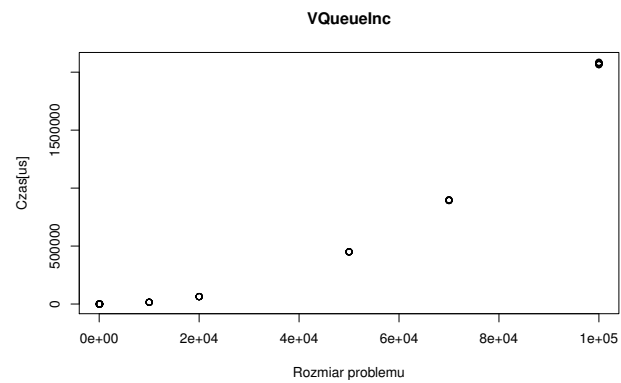
W tym dokumencie chciałbym przedstawić różnice jakie powstają pomiędzy poszczególnymi implementacjami stosu oraz kolejki na podstawie operacji dodawania.

2 Wyniki pomiarów

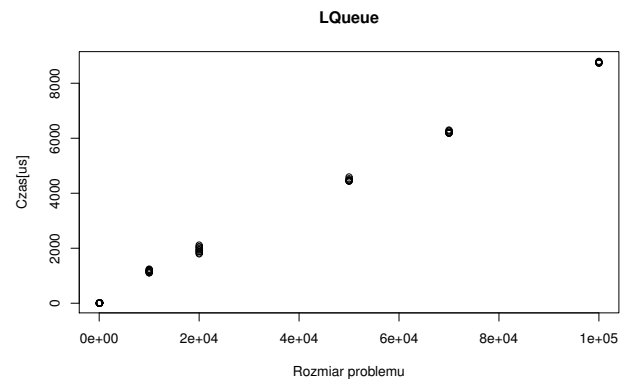
Po przeprowadzeniu testów wyniki wydają się być zgodne z intuicją. Poniżej zostały one przedstawione na wykresach czasu działania od rozmiaru problemu.

2.1 Kolejka

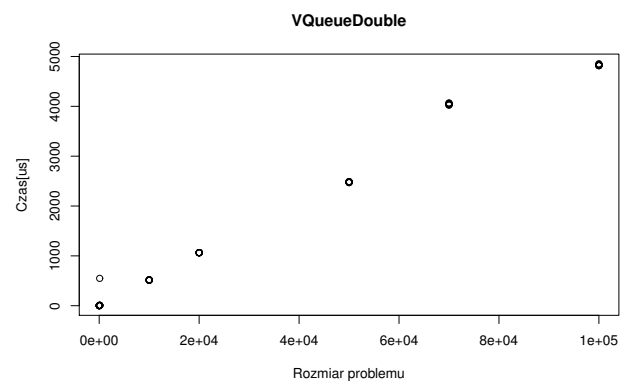
Przetestowałem kolejkę zaimplementowaną na podstawie listy oraz tablicy. W przypadku tablicy przetestowałem dwie różne metody alokacji pamięci, o których wspomniałem wcześniej. Wyniki prezentują się następująco:



Rysunek 3: Kolejka zaimplementowana na tablicy, metoda inkrementacji



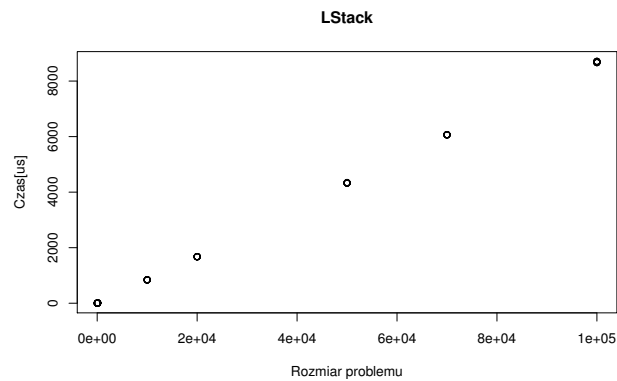
Rysunek 1: Kolejka zaimplementowana na liście



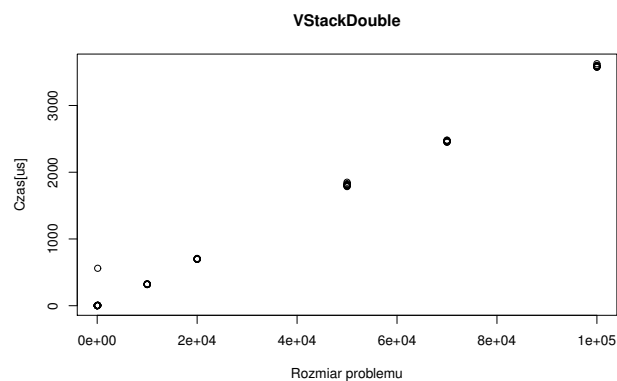
Rysunek 2: Kolejka zaimplementowana na tablicy, metoda podwajania

2.2 Stos

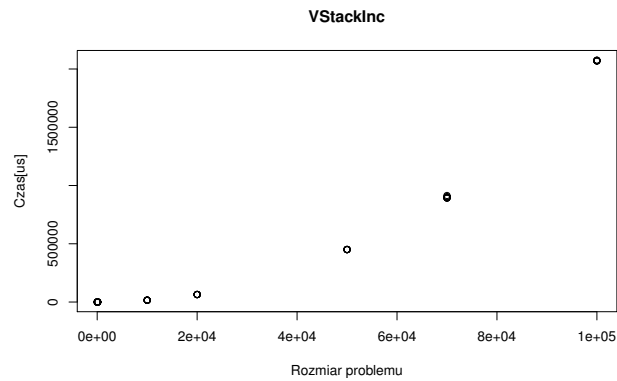
Przetestowałem stosy zaimplementowane na podstawie listy oraz tablicy. W przypadku tablicy przetestowałem dwie różne metody alokacji pamięci, o których wspomniałem wcześniej. Wyniki prezentują się następująco:



Rysunek 4: Stos zaimplementowany na liście



Rysunek 5: Stos zaimplementowany na tablicy, metoda podwajania



Rysunek 6: Stos zaimplementowany na tablicy, metoda inkrementacji

3 Wnioski

Na podstawie przeprowadzonych testów możemy wyciągnąć następujące wnioski:

- Najgorszą implementacją jest implementacja na tablicy z alokacją inkrementacyjną
- Widać, że wrzucenie n elementów na stos/kolejkę przy alokacji inkrementacyjnej wiąże się ze złożonością $\mathcal{O}(n^2)$
- Przy implementacji za pomocą listy widać, że jest to złożoność liniowa
- Alokacja, w której podwaja się dostępny rozmiar w przybliżeniu jest liniowa, chociaż można dostrzec zbiory punktów, które do tej prostej nie należą - dla takiego rozmiaru przekopiowane musiały zostać dodatkowo wszystkie elementy