Stos i Kolejka - Benchmark

Konrad Łakomy

23 03 2014

Zadanie do wykonania

1. Implementacja stosu przy pomocy:
   (a) Tablicy ver 1
   (b) Tablicy ver 2
   (c) Listy
2. Implementacja kolejki przy pomocy:
   (a) Tablicy
   (b) Listy
3. Implementacja algorytmów sortowania:
   (a) Sortowanie przez scalanie (ang. MergeSort)
   (b) Sortowanie szybkie (ang. QuickSort)

Rezultaty benchmark'u - Stos

LiczbaElementów	**QuickSort**	**MergeSort**
10	2[us]	5[us]
100	26[us]	61[us]
1000	357[us]	1005[us]

Tablica 1: Implementacja stosu w tablicy wersja 1

LiczbaElementów	**QuickSort**	**MergeSort**
10	2[us]	5[us]
100	29[us]	60[us]
1000	175[us]	1099[us]

Tablica 2: Implementacja stosu w tablicy wersja 2

LiczbaElementów	**QuickSort**	**MergeSort**
10	15[us]	22[us]
100	178[us]	331[us]
1000	1784[us]	2138[us]

Tablica 3: Implementacja stosu w liście

Rezultaty benchmark'u - Kolejka

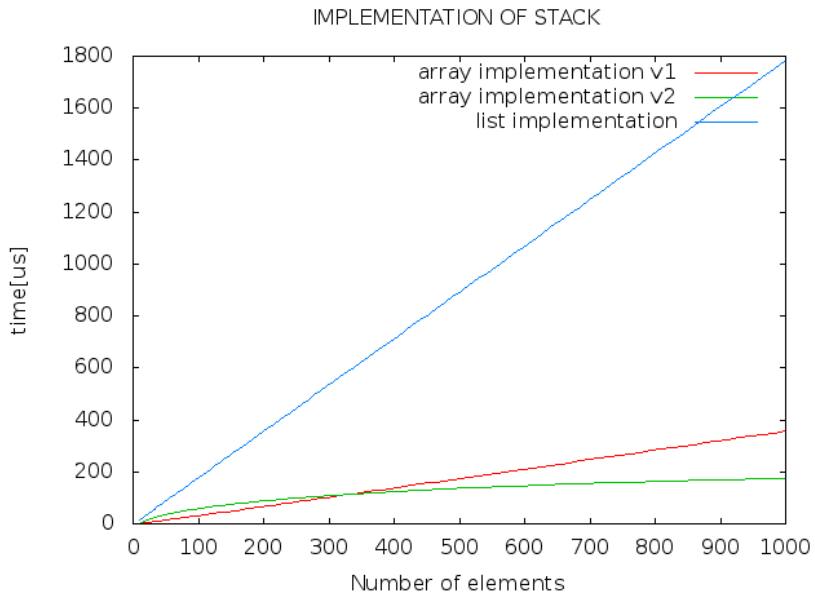
LiczbaElementów	**QuickSort**	**MergeSort**
10	2[us]	5[us]
100	28[us]	63[us]
1000	185[us]	1096[us]

Tablica 4: Implementacja stosu w kolejce wersja 1

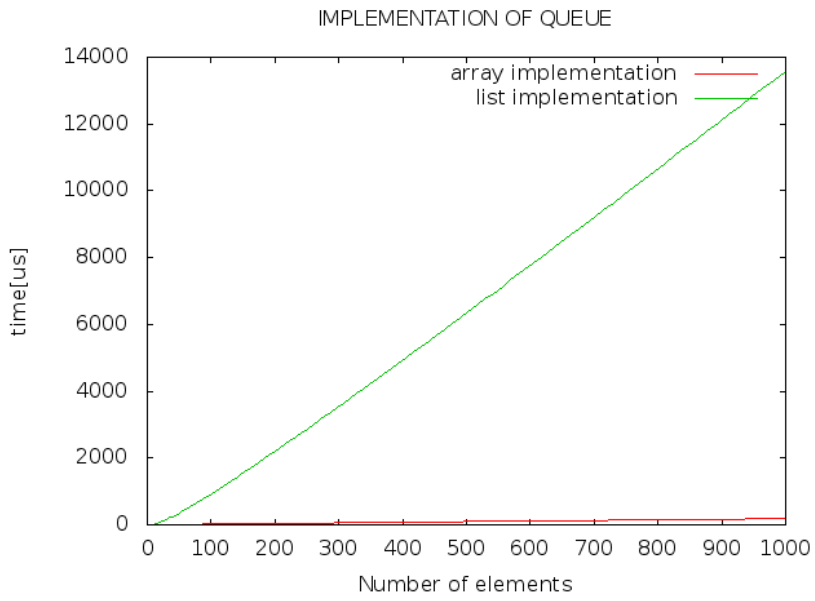
LiczbaElementów	**QuickSort**	**MergeSort**
10	14[us]	17[us]
100	367[us]	415[us]
1000	13587[us]	9027[us]

Tablica 5: Implementacja stosu w kolejce wersja 2

QuickSort - Wykresy

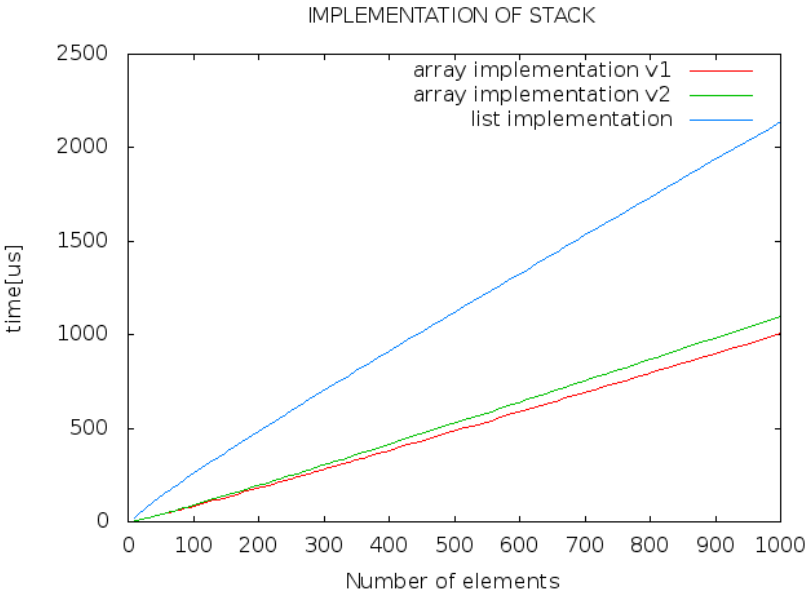


Rysunek 1: Wykres ilosci elementów od czasu wykonywania algorytmów

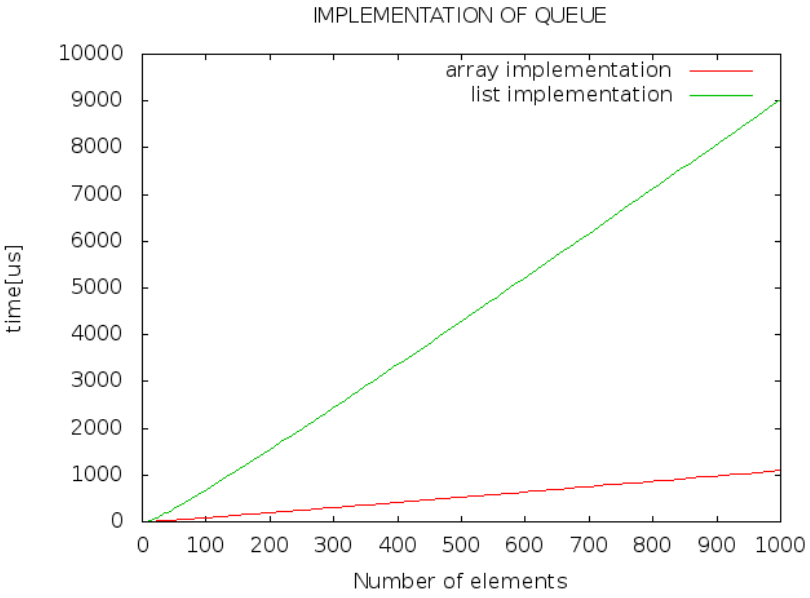


Rysunek 2: Wykres ilosci elementów od czasu wykonywania algorytmów

MergeSort - Wykresy



Rysunek 3: Wykres ilosci elementów od czasu wykonywania algorytmów



Rysunek 4: Wykres ilosci elementów od czasu wykonywania algorytmów

Wnioski

Efektywność algorytmów sortowania w dużej mierze zależy od tego w jakiej strukturze danych zaimplementowany jest dany algorytm. W zależności czy jest to stos czy kolejka oraz czy jest to implementacja w tablicy czy w liście będzie miało znaczenie przy mierzeniu czasu pracy danego algorytmu sortowania.

W przypadku obu algorytmów złożoność obliczeniowa jest równa O(n log n). Jednak algorytm sortowania szybkiego daje dużo lepsze rezultaty niż algorytm sortowania przez scalanie. W obu przypadkach korzystamy z rekurencji a wiec jest potrzebny dodatkowy czas i pamięć na jej obsłużenie.

Sortowanie stosu i kolejki zaimplementowanego na liście charakteryzuje się dłuższym czasem wykonania ponieważ sortowanie odbywa się za pomocą funkcji odpowiadających za dodawanie i usuwanie elementów z danej struktury. Inaczej jest w przypadku implementacji tablicowej gdzie sortowana jest bezpośrendio tablica na której zbudowany jest stos i kolejka.