

Laboratorium 3

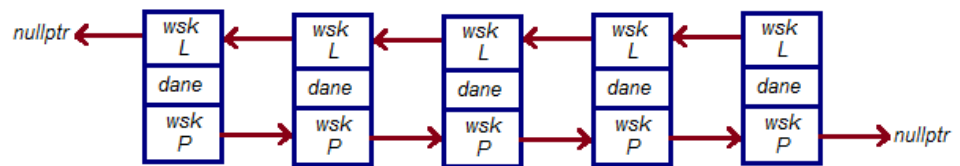
Jan Seredyński

16 kwietnia 2015

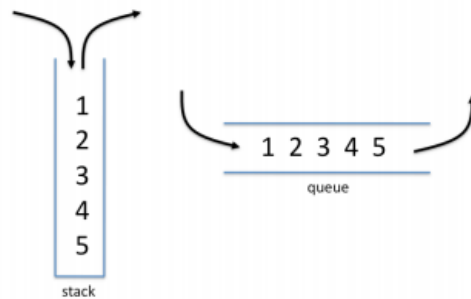
1 Wstęp

Zadaniem laboratorium jest pomiar czasu wykonania operacji wypełnienia stosu. Do wykonania analizy zastosowałem trzy implementacje: dwie tablicowe i jedna oparta na liście.

2 Schematy odpowiednich struktur



Lista dwukierunkowa



Stos, kolejka

3 Wydajność stosu na tablicy - strategia inkrementacyjna

Podczas tej próby stos jest oparty na tablicy dynamicznej, która przy każdym pushowaniu tworzy nową tablicę większą o 1, a następnie kopiuje pozostałe elementy do nowoutworzonej tablicy, a na końcu wpisuje nowy element.

Całkowity czas $T(n)$ wykonania n operacji push jest proporcjonalny do:

$$n + c + 2c + 3c + 4c + \dots + kc =$$

$$n + c(1 + 2 + 3 + \dots + k) =$$

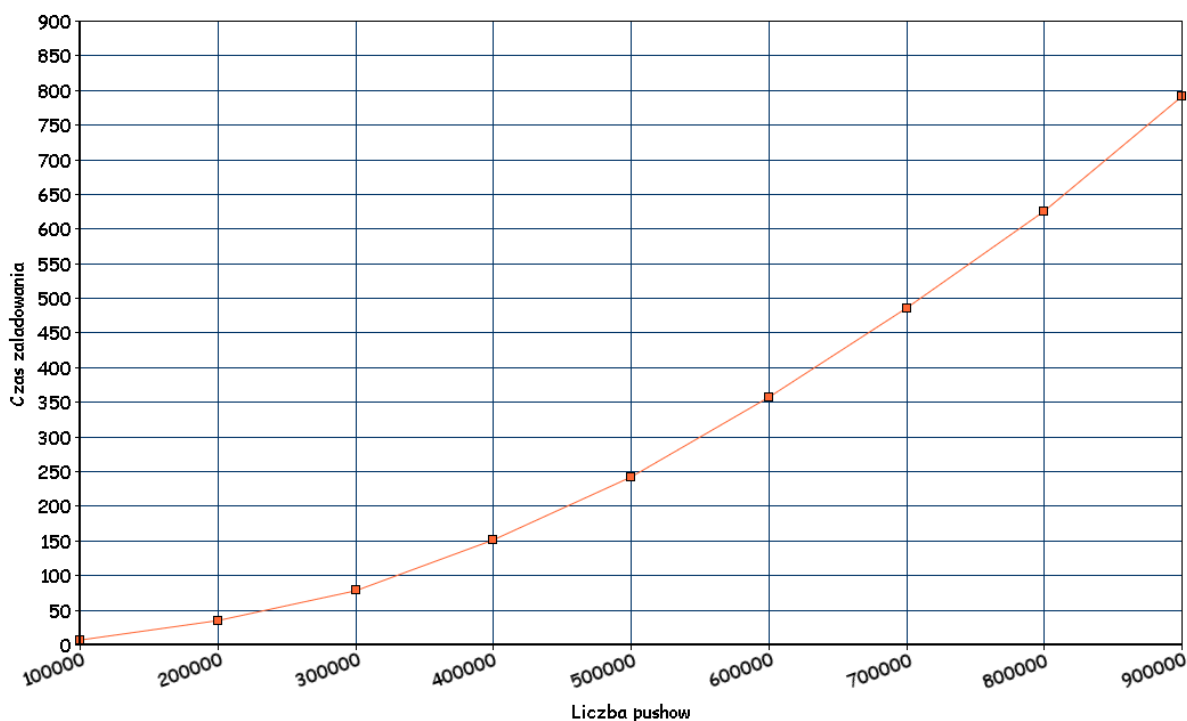
$$n + ck(k+1)/2$$

gdzie c jest stałą,

k -wielokrotność zastąpień,

$T(n)$ jest w $O(n + k^2)$, tj. $O(n^2)$

Stos alokowany o jeden element



Na

podstawie wykresu można stwierdzić, że ta implementacja ma przyrost geometryczny - kwadratowy, czyli złożoność obliczeniowa wynosi $O(n^2)$.

4 Wydajność stosu na tablicy - strategia podwajania

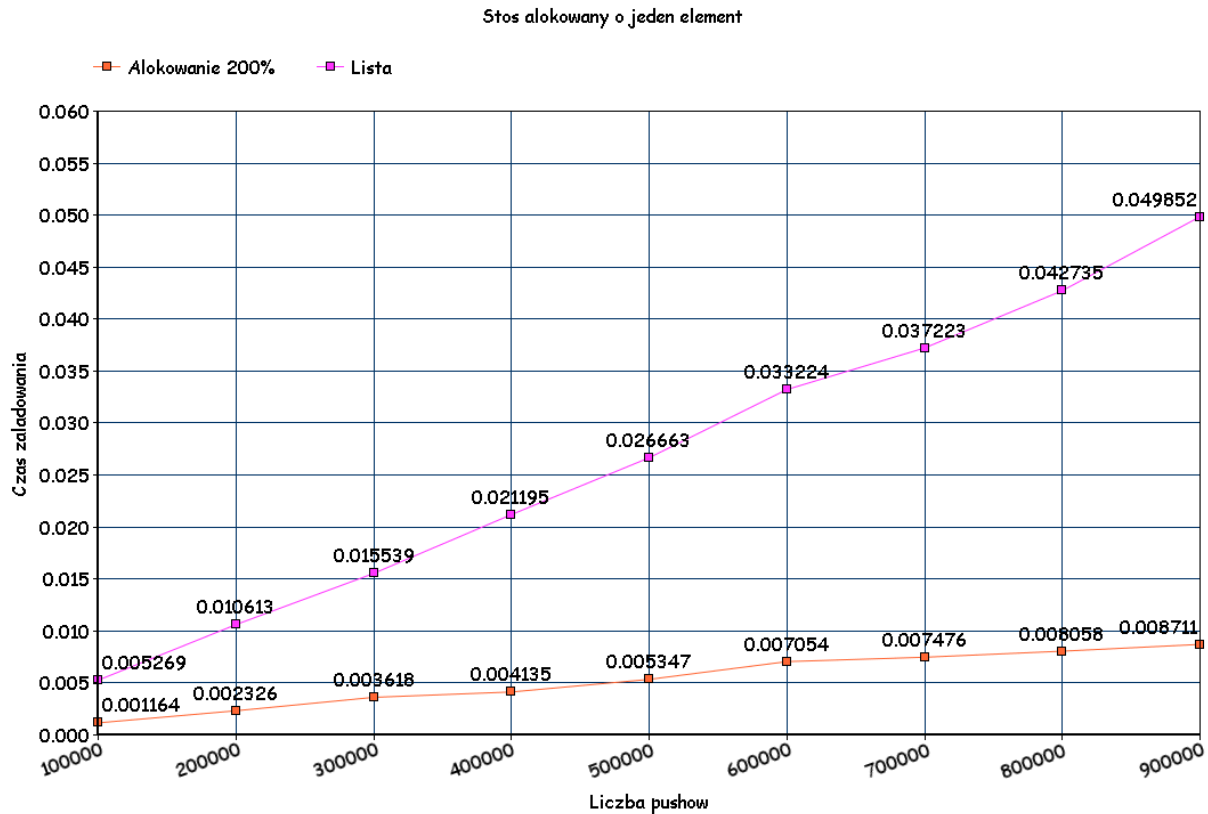
Podczas tej próby stos jest oparty na tablicy dynamicznej, która przy każdym pushowaniu sprawdza czy tablica pomieści nowy element, a gdy jest potrzeba zaalokowania nowej pamięci tworzy nową tablicę większą o 100%, a następnie kopiuje pozostałe elementy do nowoutworzonej tablicy, a na końcu pushuje nowy element.

Tablica zostanie zastąpiona $k = \log_2 n$ razy. Całkowity czas $T(n)$ jest proporcjonalny do: $n + 1 + 2 + 4 + 8 + \dots + 2^k =$

$$n + 2(k + 1) - 1 = 2n - 1$$

gdzie $T(n)$ jest w $O(n)$

Na tym samym wykresie została również złożoność obliczeniowa $O(n)$ implementacji listy.



Na podstawie wykresu można stwierdzić, że obie te implementacje mają przyrost liniowy, co spełnia założenie, że charakteryzują się złożonością obliczeniową $O(n)$.

5 Podsumowanie

Pomimo dobrze opracowanej metody pomiarowej czasu, na wykresach widać zakłócenia spowodowane pracą programów w tle.

Najbardziej wydajną implementacją jest zoptymalizowany stos na tablicy(200%), a następnie oparta na liście(obie ze złożonością obliczeniową $O(n)$). Najdłuższy czas do przeprowadzenia pushowania odnotowano przy niezoptymalizowanym stosie na tablicy, co jest spowodowane ciągłym kopiowaniem elementów do nowej powiększonej tablicy.