Laboratorium 3

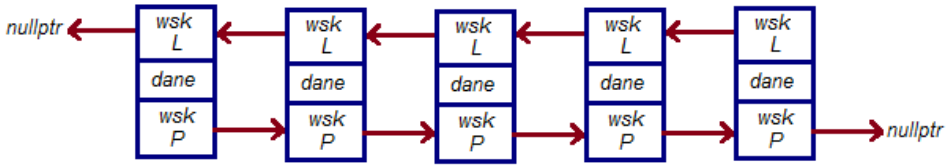
Jan Seredyński

16 kwietnia 2015

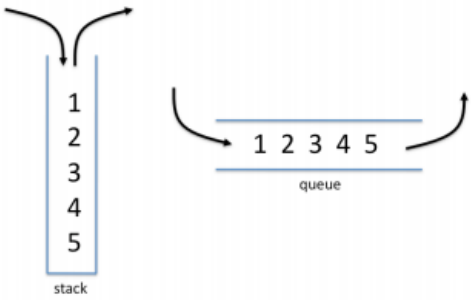
1 Wstęp

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

2 Schematy odpowiednich struktur



Lista dwukierunkowa



Stos, kolejka

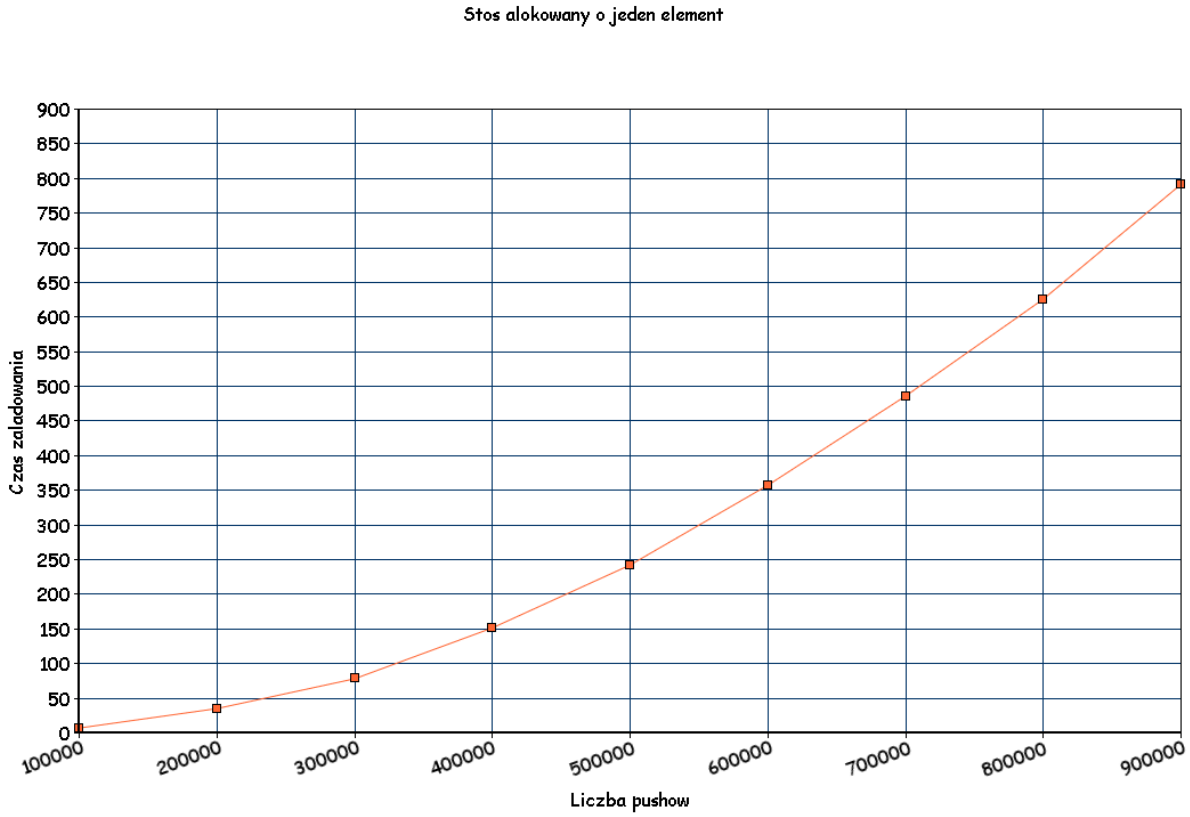
3 Wydajność stosu na tablicy - strategia inkrementacyjna

Podczas tej próby stos jest opraty na tablicy dynamicznej, która przy każdym pushowaniu twory nową tablice większą o 1, a następnie kopiuje pozostałe elementy do nowoutworznej tablicy, a na końcu wpisuje nowy element.
Całkowity czas T(n) wykonania n operacji push jest proporcjonalny do:

$$n + c + 2c + 3c + 4c + ... + kc =$$
$$n + c(1 + 2 + 3 + ... + k)=$$
$$n + ck(k+1)/2$$

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

gdzie c jest stałą,
k-wielokrotność zastąpień,



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

4 Wydajność stosu na tablicy - strategia podwajania

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

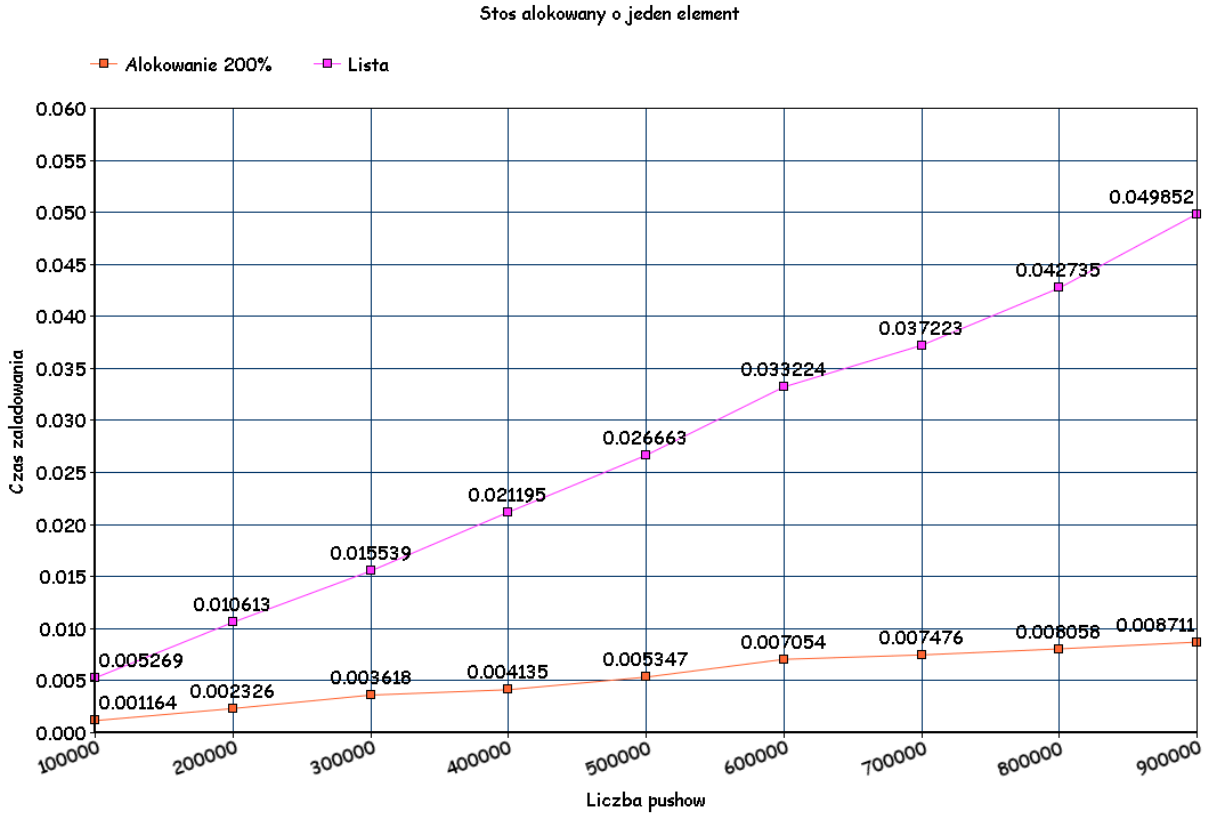
na końcu pushuje nowy element.

Tablica zostanie zastąpiona k = $log_2 n$ razy Całkowity czas T(n) jest proporcjonalny do:

$$n + 1 + 2 + 4 + 8 + ... + 2^k =$$
$$n + 2^(k + 1) - 1 = 2n - 1$$

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 spełdnia założenie, że charakteryzują się złożonością obliczniową O(n).

5 Podsumowanie

Najbardziej wydajnyą implementacją jest zoptymalizowanystos na tablicy(200%). Optymalizacje tutaj otrzymujemy poprzez strategie podwajania, która osiąga złożoność obliczeniową O(n), co jest dużą różnicą w porównaniu do strategi inkrementacyjnej której wyniki złożoności obliczeniowej sięgają aż $O(n^2)$.

Implementacja oprta na liście wydajnym algorytmem, ponieważ jej złożoność obliczeniowa wynosi O(n), którą osiągamy dzięki temu, że nie ma potrzeby kopiowania starych elementów przy tworzeniu nowej większej tablicy, jak w przypadku wcześniejszej implementacji.