

Zadanie projektowe nr 1

Badanie efektywności operacji na danych w podstawowych strukturach danych

Należy zaimplementować oraz dokonać pomiaru czasu działania operacji takich jak dodawanie elementu, usunięcie elementu i wyszukanie elementu w następujących strukturach danych:

- a) tablica,
- b) lista dwukierunkowa,
- c) kopiec binarny (typu maksimum – element maksymalny w korzeniu) ,
- d) drzewo czerwono-czarne.

Należy przyjąć następujące założenia:

- podstawowym elementem struktur jest 4 bajtowa liczba całkowita ze znakiem,
- wszystkie struktury danych powinny być alokowane dynamicznie (w przypadku tablic powinny zajmować możliwie jak najmniej miejsca tzn. powinny być relokowane przy dodawaniu/usuwaniu elementów),
- w przypadku tablicy i listy rozpatrzyć osobno operacje dodawania i usuwania elementu na następujących pozycjach:
 - a. początek tablicy/listy,
 - b. koniec tablicy/listy,
 - c. dowolne miejsce tablicy/listy (określane jako parametr odpowiedniej funkcji).
- należy zmierzyć czasy wykonywania poszczególnych operacji w funkcji rozmiaru danej struktury (liczby elementów w niej przechowywanych). Ponieważ pojedynczy pomiar może być obarczony znacznym błędem oraz otrzymane wyniki mogą zależeć także od rozkładu danych, to pomiary dla konkretnego rozmiaru struktury należy wykonać wielokrotnie (np. 100 razy – za każdym razem generując nowy zestaw danych), a wyniki uśrednić. Liczbę przechowywanych elementów należy dobrać eksperymentalnie (np. może to być 1000, 2000, 5000, 10000, 20000) w zależności od wydajności sprzętu. Czasu pomiaru powinien uwzględniać wyłącznie czas wykonania danej operacji (nie powinien uwzględniać p. czasu generacji danych).
- należy mieć na uwadze, że czas wykonywania operacji może zależeć od wartości przechowywanych elementów, co należy uwzględnić w pomiarach i wnioskach,
- dodatkową funkcją programu musi być możliwość sprawdzenia poprawności zaimplementowanych operacji i zbudowanej struktury danych,

- do dokładnego pomiaru czasu w systemie Windows w c++ można skorzystać z funkcji `QueryPerformanceCounter` lub `std::chrono::high_resolution_clock` (opis na stronie <http://cpp0x.pl/forum/temat/?id=21331>,
- dopuszczalnymi językami programowania są języki kompilowane do kodu natywnego (np. C, C++), a nie interpretowane lub uruchamiane na maszynach wirtualnych (np. JAVA, .NET, Python); dopuszczalne jest odstępstwo od tej reguły za zgodą prowadzącego,
- używanie okienek nie jest konieczne i nie wpływa na ocenę (wystarczy wersja konsolowa),
- nie wolno korzystać z gotowych bibliotek np. STL, Boost lub innych – wszystkie algorytmy i struktury muszą być zaimplementowane przez studenta (nie kopiować gotowych rozwiązań),
- realizacja zadania powinna być wykonana w formie jednego programu,
- kod źródłowy powinien być komentowany,
- program musi skompilowany do wersji exe (i w takiej wersji zostanie poddany testom).

Sprawdzenie poprawności zbudowanej struktury/operacji obejmuje:

- utworzenie struktury na podstawie danych zapisanych w pliku tekstowym. Pierwsza liczba określa rozmiar struktury, następnie należy odpowiednią liczbę danych np. każda liczba w osobnej linii,
- wyświetlenie struktury na ekranie (w przypadku drzew zaproponować odpowiednią formę, która uwzględni relacje między elementami tej struktury),
- możliwość wykonania wszystkich przewidzianych operacji na danej strukturze (wybór operacji najlepiej zrealizować w formie menu).

Sprawozdanie (w formie papierowej) powinno zawierać:

- krótki wstęp w którym zostaną przedstawione złożoności obliczeniowe operacji w implementowanych strukturach na podstawie literatury,
- plan eksperymentu, czyli założenia co do wielkości struktur, sposobu generowania elementów tych struktur, sposobie pomiaru czasu, itp.
- zestawienie wyników w formie tabelarycznej i graficznej (czas wykonania operacji w funkcji liczby elementów); wyniki należy przedstawić osobno dla poszczególnych operacji,
- wnioski dotyczące efektywności poszczególnych struktur w zależności od zastosowań, wielkości struktury itp., wskazać (jeśli są) przyczyny rozbieżności pomiędzy uzyskanymi eksperymentalnie złożonościami, a teoretycznymi,
- załączony kod źródłowy w formie elektronicznej (skopiować cały projekt oraz wersję skompilowaną programu),
- jednostki w sprawozdaniu powinny być mianowane i mieć zachowaną odpowiednią precyzję.

Ocena projektu:

3.0 – eksperymenty na tablicy, liście i kopcu binarnym

3.5 – eksperymenty na tablicy, liście i kopcu binarnym (program w wersji obiektowej)

4.0 – eksperymenty na tablicy, liście i kopcu binarnym i drzewie czerwono-czarnym

5.0 – eksperymenty na tablicy, liście i kopcu binarnym i drzewie czerwono-czarnym (program w wersji obiektowej)

5,5 – eksperymenty na tablicy, liście i kopcu binarnym i drzewie czerwono- czarnym oraz dodatkowo drzewie AVL (program w wersji obiektowej)