

Zadanie projektowe nr 1

Badanie efektywności operacji dodawania (wstawiania), usuwania oraz wyszukiwania elementów 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:

- ❖ tablica,
- ❖ lista dwukierunkowa,
- ❖ kopiec binarny,
- ❖ drzewo BST (-/+DSW)
- ❖ drzewo czerwono-czarne.

1. Założenia

- ❖ podstawowym elementem struktur, kluczem K , jest 4 bajtowa liczba całkowita ze znakiem,
- ❖ wszystkie struktury danych powinny być alokowane dynamicznie; w przypadku tablic powinny zajmować jak najmniej miejsca – powinny być relokowane przy dodawaniu/usuwaniu kluczy,
- ❖ w przypadku tablicy i listy rozpatrzyć osobno operacje dodawania i usuwania klucza na początku, końcu i w wybranym miejscu (szczegóły w pkt. 2).
- ❖ należy zmierzyć czasy wykonywania poszczególnych operacji w funkcji rozmiaru danej struktury (liczby kluczy w niej przechowywanych). W przypadku zbyt krótkich czasów można albo zwiększyć liczbę danych pomiarowych, albo powtórzyć pomiar np. dziesięciokrotnie. Ponieważ wartość pomiaru zależy również od rozkładu danych (wartości liczb) należy przygotować zestawy zawierające różną liczbę danych o różnych wartościach. Należy określić maksymalną, możliwą do umieszczenia w dostępnej pamięci ($MaxMem$), liczbę danych w strukturze $MaxD = MaxMem/K$.

Na tej podstawie stworzyć zestawy (w sumie) danych:

- o małej ($10 - 50$),
- średniej ($\frac{1}{2} MaxD$)
- dużej ($MaxD$) liczbie danych

zawierających liczby z zakresów:

- $[10, 100]$,
- $\sim \frac{1}{2} MaxMem$ oraz
- $\sim MaxMem$ każda.

- ❖ należy mieć na uwadze, że czas wykonywania operacji może zależeć od wartości przechowywanych kluczy (patrz wyżej), co należy uwzględnić w pomiarach i wnioskach,
- ❖ funkcją programu musi być również możliwość sprawdzenia poprawności zaimplementowanych operacji i zbudowanej struktury (szerzej w na ten temat w dalszej części dokumentu),
- ❖ wygląd interfejsu użytkownika oraz sposobu obsługi programu (wersja konsolowa) pokazane są w formie kodu programu, który znajduje się na stronie

<http://jaroslaw.mierzwa.staff.iiar.pwr.edu.pl/sdizo/menu.cpp>

w przypadku wykonania wersji okienkowej funkcjonalność programu nie może być zmniejszona (program musi być co najmniej tak funkcjonalny jak wersja konsolowa),

- ❖ dopuszczalnymi językami programowania są języki kompilowane do kodu natywnego (np. C, C++), a nie interpretowane lub uruchamiane na maszynach wirtualnych (np. JAVA, .NET, Phyton),
- ❖ dokładnego pomiaru czasu w systemie Windows dokonujemy z wykorzystaniem funkcji *QueryPerformanceCounter* lub `std::chrono::high_resolution_clock`. Opis na stronie:

<http://cpp0x.pl/forum/temat/?id=21331>,

- ❖ nie wolno korzystać z gotowych bibliotek np. STL, Boost lub innych – wszystkie algorytmy i struktury muszą być zaimplementowane (nie kopiować gotowych rozwiązań),
- ❖ kod źródłowy powinien być komentowany.

Badanie muszą zostać wykonane na przygotowanych, zgodnie z powyższymi założeniami, zestawach oraz na kilku (kilkunastu) instancjach losowych. Należy sprawdzić czy dla losowo wygenerowanych instancji złożoności czasowe i pamięciowe wykonywania operacji dodawania (wstawiania), usuwania i wyszukiwania są takie jak dla danych testowych, a jeśli od nich odbiegają, to dlaczego?

2. Sprawdzenie poprawności zbudowanych struktur/operacji:

- ❖ utworzenie struktury z liczb zapisanych w pliku tekstowym. Każda liczba będzie w osobnej linii, natomiast pierwsza liczba określa ilość zapisanych liczb w pliku; nazwa pliku nie może być zapisana w programie,
- ❖ wyświetlenie struktury na ekranie:
 - zawartość tablicy i listy należy wyświetlać rozdzielając klucze znakami odstępu (spacja); do wyświetlania drzew można skorzystać z gotowych rozwiązań,
 - zawartość struktury musi być automatycznie wyświetlana po wykonaniu operacji; przy czym zawartość przed operacją ma również być widoczna na ekranie.
- ❖ możliwość wykonania operacji na strukturze z tym, że w przypadku:
 - dodanie/wstawianie do tablicy – podawane są klucz i pozycja (indeks), którą należy wstawić na podaną pozycję; jeżeli indeks jest mniejszy niż pierwszy lub większy niż

ostatni znany, należy rozszerzyć tablicę (utworzyć odpowiednią liczbę pól) i wstawić podany klucz, a miejsca pomiędzy uzupełnić kluczem oznaczającym miejsce puste; jeżeli podany indeks istnieje już w tablicy, należy w jego miejsce wstawić podany klucz a pozostałe klucze przesunąć (w efekcie tablica będzie dłuższa o jedną pozycję),

- dodanie/wstawianie do listy – podawane są klucz i pozycja (numer pola na liście) reszta jak wyżej,
- dodanie/wstawienie do kopca binarnego - podawany jest indeks oraz klucz; jeżeli indeks jest większy od największego istniejącego w strukturze, wtedy indeks ma wartość o jeden większą od największej do tej pory; w przeciwnym przypadku należy pod wskazany indeks wstawić nowy klucza pozostałe (większe) indeksy inkrementować nie zmieniając wartości kluczy;
- dodanie/wstawienie do drzewa BST i czerwono-czarnego – tak samo jak w przypadku kopca binarnego,
- usunięcie z tablicy – podawane są pozycja (indeks) i/lub klucz do usunięcia,
- usunięcie z listy – podawane są pozycja (numer pola) i/lub klucz do usunięcia,
- usunięcie z kopca binarnego – podawane są (indeks) i/lub klucz do usunięcia,
- usunięcie z drzewa BST i czerwono-czarnego – podawane są (indeks) i/lub klucz do usunięcia,
- wyszukiwania (dla wszystkich struktur) – zostanie podana liczba, którą należy odszukać – należy wyświetlić pozycję(e) (indeks lub numer pola) oraz klucz(e).
- dla drzewa BST z algorytmem DSW – algorytm DSW wywoływany (uruchamiany) jest automatycznie po dodaniu lub usunięciu klucza zatem czasy tych operacji mierzymy łącznie z czasem wykonania równoważenia.

Należy zmierzyć czasy trwania poszczególnych operacji w funkcji rozmiaru instancji (liczba elementów przechowywanych). W przypadku uzyskiwania zbyt krótkich czasów działania pomiar należy powtórzyć np. 10-krotnie, a następnie wyliczyć czas trwania pojedynczej operacji.

Wyświetlana zawartość struktur musi być czytelna. Zawartość tablicy i listy należy wyświetlać rozdzielając klucze znakami odstępu (spacja). Do wyświetlania drzew można skorzystać z gotowych rozwiązań.

Zawartość struktury musi być automatycznie wyświetlana po wykonaniu operacji. Przy czym zawartość przed operacją ma również być widoczna na ekranie.

3. Sprawozdanie

- ❖ opis zadania z definicjami (charakterystyką) badanych struktur, definicją efektywności badanych procedur oraz oszacowaniem (teoretycznym) złożoności poszczególnych operacji (wraz z uzasadnieniem) - wszystko w oparciu o literaturę (podstawowa podana na końcu), ale możliwie własnymi słowami; plagiaty będą surowo karane !!!
- ❖ plan eksperymentu, czyli jakie struktury i jak będą badane; założenia co do sposobu generowania elementów tych struktur lub ich źródeł (należy pamiętać, aby spełniały podane wcześniej założenia), sposobie pomiaru czasu; należy pamiętać, że w każdym badaniu można

badać, np. na czas wykonania operacji, wpływ tylko jednej zmiennej (np. dodanie elementu dla małej struktury, a nie dodanie elementu dla dowolnie dużej struktury z późniejszym "filtrowaniem" wielkości i wyciąganiem wniosków, co do wpływu na czas wykonania operacji, dla małych, średnich i dużych struktur (co do wielkości)),

- ❖ zestawienie wyników w formie tabelarycznej i graficznej - wyniki osobno dla poszczególnych operacji; wykresy osobne dla poszczególnych operacji oraz zbiorcze np. dla czasu wykonania w zależności od wielkości struktury albo czasu w zależności od rodzaju operacji dla wszystkich struktur; na wykresach (w przypadku oszacowań złożoności czasowych) umieszczać linie trendu wynikające z pomiarów oraz linie odpowiadające teoretycznym oszacowaniom pesymistycznym i średnim,
- ❖ oszacowania, na podstawie wyników, złożoności pesymistycznych i średnich wszystkich operacji; oszacowanie musi posiadać potwierdzenie w uzyskanych wynikach; w przypadku rozbieżności (zwłaszcza znaczących: teoria mówi $O(n)$, wyniki pokazują $O(n^2)$)
- ❖ wnioski płynące z wykonanych eksperymentów dotyczące złożoności czasowej i pamięciowej operacji na badanych strukturach danych; w przypadku badania czasów operacji usuwania elementów należy uwzględnić (odjąć) również czas relokacji elementów tablicy.

4. Pytania dodatkowe, czyli co należałoby wiedzieć po zakończeniu

- ❖ Czy w tablicy mogą występować puste, czyli bez klucza podanego przez użytkownika i do czego może służyć ich istnienie?
- ❖ Czy w pozostałych strukturach mogą występować miejsca puste? Odpowiedź uzasadnić?
- ❖ Czy samodzielnie budowane struktury są/mogą być efektywniejsze od implementowanych w dostępnych bibliotekach? Odpowiedź uzasadnić.

5. Ocena projektu

- 3,0 – badanie operacji na tablicy, liście i kopcu binarnym
- 3,5 – badanie operacji na tablicy, liście i kopcu binarnym (program w wersji obiektowej)
- 4,0 – badanie operacji na tablicy, liście, kopcu binarnym i drzewie BST (bez algorytmu równoważenia DSW)
- 4,5 – badanie operacji na tablicy, liście, kopcu binarnym i drzewie BST (z algorytmem równoważenia DSW)
- 5,0 – badanie operacji na tablicy, liście, kopcu binarnym i drzewie czerwono-czarnym
- 5,5 – badanie operacji na tablicy, liście, kopcu binarnym i drzewie czerwono-czarnym, wykonanie programu w wersji obiektowej oraz implementacja i badanie złożoności operacji dodawanie (wstawiania), usuwania i wyszukiwania w drzewie AVL

Od oceny 4,0 wszystkie programy muszą być zrealizowane w wersji obiektowej.

6. Termin złożenia projektu

Zadanie projektowe nr 1 należy złożyć do dnia **04 kwietnia 2017r.** Każdy rozpoczęty tydzień spóźnienia oznacza obniżenie oceny uzyskanej z zadania projektowego o jeden stopień. Należy również pamiętać, że ocena końcowa z projektu (jako formy zajęć) liczona będzie jako średnia z sumy ocen z poszczególnych zadań projektowych z uwzględnieniem spóźnień. Oznacza to, że

aby uzyskać ocenę dostateczną z projektu (formy zajęć) suma z trzech zadań projektowych z uwzględnieniem ewentualnych spóźnień, musi wynieść co najmniej 9,0. W przeciwnym przypadku projekt (forma zajęć) zostanie oceniony na ocenę niedostateczną.

7. Forma złożenia projektu

Patrz zjo.iia.pwr.wroc.pl/sdizo.html **składanie projektów** (pdf)

8. Źródła

[1] T. H. Cormen (i inni) , Wprowadzenie do algorytmów, WNT, Warszawa, 1997

[2] A. Drozdek, C++ Algorytmy i struktury danych, Helion, Gliwice, 2001

[3] http://eduinf.waw.pl/inf/alg/001_search/index.php

Plagiaty będą surowo karane !!!