

Zadanie laboratoryjne nr 2

Badanie efektywności operacji dodawania, usuwania oraz wyszukiwania elementów w drzewiastych strukturach danych

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

- ❖ kopiec binarny,
- ❖ drzewo BST,
- ❖ 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,
- ❖ 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),

- ❖ dopuszczalnymi językami programowania są języki kompilowane do kodu natywnego (np. C, C++), a nie interpretowane lub uruchamiane na maszynach wirtualnych (np. JAVA, .NET, Python),
- ❖ sposoby dokładnego pomiaru czasu w systemie Windows podano na stronie:
http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pamsi/debug_and_time.pdf,
- ❖ 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 liczbę zapisanych liczb w pliku,
- ❖ wyświetlenie tej struktury na ekranie,
 - do wyświetlania drzew można skorzystać z gotowych rozwiązań, np. z [3],
 - zawartość struktury musi być automatycznie wyświetlana po wykonaniu operacji; przy czym zawartość przed operacją ma być również widoczna na ekranie.
- ❖ możliwość wykonania operacji na strukturze z tym, że w przypadku:
 - 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 klucz a pozostałe (większe) indeksy inkrementować nie zmieniając wartości kluczy;
 - dodanie/wstawienie do drzewa BST oraz drzewa czerwono-czarnego – tak samo jak w przypadku kopca binarnego,
 - usunięcie z kopca binarnego, drzewa BST oraz drzewa 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 uruchamiany jest automatycznie po dodaniu nowego lub usunięciu klucza, zatem czasy wykonania tych operacji należy mierzyć łącznie z czasem równoważenia.

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. Ocena projektu

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

5. 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