

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

**Zarządzanie w systemach i sieciach
komputerowych - projekt**

**Ocena wydajności implementacji algorytmu
przeszukiwania drzewa BST z wykorzystaniem technologii
C++ i Python.**

AUTORZY:

Gracjan Serwinowski

Indeks: 264166

E-mail: 264166@student.pwr.edu.pl

Miłosz Jaszewski

Indeks: 259093

E-mail: 259093@student.pwr.edu.pl

PROWADZĄCY ZAJĘCIA:

Dr inż. Robert Wójcik, K30W04D03

OCENA PRACY:

Wrocław, 2024

Opis założeń projektu

1. Cel i zakres projektu

Celem niniejszego projektu jest przeprowadzenie szczegółowych badań nad efektywnością implementowanych algorytmów operujących na różnych strukturach danych. Analiza ta będzie obejmować zarówno złożoność teoretyczną algorytmów, jak i ich praktyczną wydajność, mierzoną czasem wykonania operacji. Projekt zakłada porównanie dwóch wersji programów realizujących te same operacje: jednej napisanej w języku Python oraz drugiej w C++. Kluczowe operacje, które będą badane, obejmują wstawianie, usuwanie i wyszukiwanie elementów w dynamicznych strukturach danych, takich jak tablica, lista liniowa oraz drzewo BST.

Zakres pracy obejmuje:

- Analizę literaturową dotyczącą złożoności obliczeniowej oraz wydajności różnych struktur danych.
- Implementację algorytmów i struktur danych w dwóch wybranych językach programowania (Python i C++).
- Przeprowadzenie serii testów mających na celu ocenę efektywności algorytmów.
- Zbieranie i analiza wyników, w tym wizualizacja danych przy pomocy wykresów.
- Sformułowanie wniosków na podstawie przeprowadzonych badań.

2. Sformułowanie problemu

Problem analizy wydajności algorytmów zostanie rozwiązany przy uwzględnieniu następujących założeń:

- Elementami struktur danych będą liczby całkowite, co uprości analizę i umożliwi porównywanie wyników.
- Struktury danych muszą być dynamiczne, co oznacza, że będą musiały być zaimplementowane z użyciem wskaźników, co jest szczególnie istotne dla list i drzew.
- Badania będą obejmować różne scenariusze dodawania elementów, takie jak:

- Dodawanie na początek, koniec i w losowe miejsce w tablicach oraz listach.
- Wstawianie elementów w drzewie BST zgodnie z różnymi algorytmami, w tym dla układów zdegenerowanych do listy liniowej oraz różnorodnych typów danych (minimalne, maksymalne, losowe).

3. Analiza złożoności obliczeniowej problemu

Na obecnym etapie nie można dokładnie określić złożoności obliczeniowej rozwiązywanego problemu, jednak spodziewane są różnice w czasie działania algorytmów w zależności od struktury danych oraz rozkładu danych wejściowych. W szczególności, złożoność obliczeniowa operacji wstawiania, usuwania i wyszukiwania dla każdej z wybranych struktur danych zostanie poddana dokładnej analizie. Rozważane będą różne przypadki (najlepszy, średni i najgorszy), co pozwoli na uzyskanie pełniejszego obrazu efektywności implementacji.

4. Metoda i algorytmy rozwiązywania problemu

Analiza wydajności będzie przeprowadzona z wykorzystaniem zarówno podejść teoretycznych, jak i praktycznych. Dla problemu decyzyjnego wykorzystane zostaną standardowe metody analizy algorytmicznej oraz porównania czasów działania. Dla problemu optymalizacyjnego będą testowane różne algorytmy, takie jak:

- Algorytmy klasyczne wstawiania i usuwania dla każdej z struktur.
- Różnorodne podejścia do wyszukiwania (np. wyszukiwanie binarne dla BST, liniowe dla tablicy i listy).
- Optymalizacje, takie jak zrównoleglanie operacji oraz wykorzystanie odpowiednich bibliotek do pomiarów czasu.

5. Technologie i narzędzia implementacji

Program zostanie napisany w dwóch językach: C++ i Python. Do implementacji w C++ wykorzystane zostaną biblioteki STL, co umożliwi korzystanie z wydajnych struktur danych i algorytmów. Projekt będzie realizowany w środowisku Microsoft Visual Studio dla C++ oraz PyCharm lub VS Code dla Pythona.

6. Testowanie i ocena jakości rozwiązań

Skonfigurowany system będzie poddany rygorystycznym testom mającym na celu weryfikację poprawności działania oraz efektywności algorytmów. Planowane są testy jednostkowe, akceptacyjne oraz porównawcze. Czas działania algorytmów będzie mierzony za pomocą odpowiednich funkcji do pomiaru czasu w obu językach, a wyniki zostaną zanalizowane i zestawione w formie wykresów. Istotne informacje będą zapisywane do plików tekstowych, a wizualizacja wyników pomoże w interpretacji uzyskanych danych.

Literatura

1. Cormen T., Leiseron C., Rivest R., *Wprowadzenie do algorytmów*, WNT, Warszawa, 2001.
 2. Błażewicz J., *Problemy optymalizacji kombinatorycznej*, PWN, Warszawa, 1996.
 3. Stadnicki J., *Teoria i Praktyka Rozwiązywania Zadań Optymalizacji z przykładami zastosowań technicznych*, PWN, Warszawa, 2017.
 4. Strona internetowa: <https://en.cppreference.com/w/cpp/chrono>, 12.10.2020.
 5. ChatGPT, <https://openai.com/chatgpt>, 02.10.2024.
 6. Strona AI, <https://hix.ai/pl/hub/writing/best-chatgpt-alternatives>, 07.10.2024.
-

