Konspekt projektu z przedmiotu Organizacja i Architektura Komputerów

Temat: Testowanie pierwszości.

Cel projektu: Implementacja algorytmów pozwalających na określenie, czy dana liczba jest pierwsza czy złożona oraz ocena jakości otrzymanych wyników pod kątem czasu pracy i poprawności rezultatów.

Główne założenia: W ramach projektu postaram się zaimplementować następujące algorytmy:

* Metoda naiwna (sprawdzenie podzielności polegająca na dzieleniu zadanej liczby n przez 2,3,…, n-1. Możliwe modyfikacje np. dzielenie liczb tylko do ).
* Test pierwszości Fermata.
* Test pierwszości Solovaya-Strassena lub/i Millera-Rabina.
* Test Agrawal-Kayal-Saxena AKS.
* Możliwe dodatkowe takie jak: test Frobeniusa (połączenie testu Millera-Rabina i Solovaya-Strassena), Baillie-PSW (połączenie testu Fermata i Milllera-Rabina wraz z liczbami pseudo pierwszymi Lucasa).

Następnie na podstawie wybranej populacji danych wejściowych porównywane będą czasy działania poszczególnych algorytmów oraz jakość otrzymanych rozwiązań w postaci liczby poprawnych rozwiązań w stosunku do rozmiaru populacji. Do weryfikacji czy otrzymana liczba jest liczbą pierwszą zostaną wykorzystane dostępne w literaturze i internecie spisy liczb pierwszych.

Projekt zostanie zaimplementowany w języku Python.

Główne ograniczenia:

* Brak możliwości podania nieskończenie dużej liczby do przeprowadzenia testu pierwszości.
* Ograniczona prędkość wykonywania obliczeń wynikająca z ograniczeń sprzętowych komputera oraz specyfikacji języka Python (m.in. interpretowany kod, występowanie maszyny wirtualnej).
* Należy przeprowadzić kontrolę, by podana przez użytkownika liczba należała do zbioru liczb całkowitych dodatnich.

Możliwe problemy do napotkania: Możliwe problemy do napotkania przy implementacji to m.in.:

* Test pierwszości Fermata nie gwarantuje poprawności otrzymanych wyników.   
  Istnieje pewien zbiór liczb złożonych zwanych liczbami Carmichaela, które są błędnie uznawane za liczby pierwsze.
* Testy Solovaya-Strassena i Millera-Rabina są testami probabilistycznymi. Oznacza to, że poprawność otrzymanych wyników musi być weryfikowana, a także należy dobrać takie współczynniki by zmaksymalizować prawdopodobieństwo poprawnej weryfikacji liczby pierwszej.
* Podanie przez użytkownika zbyt dużej liczby wejściowej. Jednym z rozwiązań jest ograniczenie liczby znaków danej wejściowej.

Harmonogram pracy:   
Gotowy projekt powinien zostać wykonany w przeciągu 12 tygodni. Zakładam, że harmonogram prac będzie wyglądał jak prezentowany poniżej.

* Tydzień (1-2): Przygotowanie konspektu pracy, wybranie tematu projektu i zaplanowanie pracy.
* Tydzień (3-4): Implementacja funkcji pomocniczych takich jak: funkcja weryfikująca liczbę pierwszą z zapisaną liczbą na liście liczb pierwszych, funkcja mierząca czas wykonywania algorytmów, funkcja zapisująca otrzymane wyniki do zewnętrznych plików, funkcje przyjmujące wyniki i tworzące na ich podstawie wykresy itp. Oraz implementacja metody naiwnej.
* Tydzień (5): Implementacja testu pierwszości Fermata.
* Tydzień (6-7): Implementacja testu pierwszości Solovaya-Strassena lub/i Millera-Rabina.
* Tydzień (8): Implementacja testu Agrawala-Kayala-Saxena.
* Tydzień (9-10): Zebranie wyników algorytmów, przeprowadzenie testów i oceny jakości zaimplementowanych algorytmów, wprowadzenie ew. poprawek, przygotowanie materiałów do stworzenia raportu.
* Tydzień (11): Tworzenie sprawozdania z projektu.
* Tydzień (12): Oddanie gotowego projektu w tym sprawozdanie oraz kod programu.