

Zadanie 1

Napisać program wyznaczający przybliżenie liczby π . W tym celu należy wykorzystać równość:

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

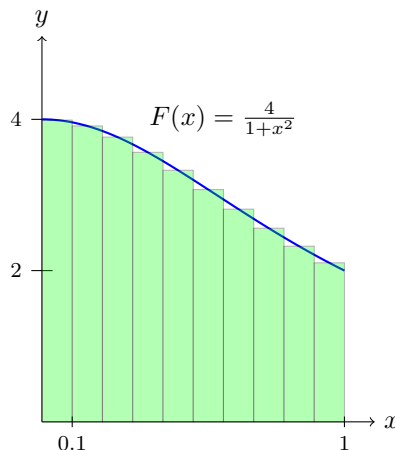
Całka w powyższej formule może zostać przybliżona za pomocą sumy pól prostokątów:

$$\sum_{i=1}^N F(x_i) \Delta x \approx \pi$$

gdzie każdy prostokąt ma szerokość $\Delta x = \frac{1-0}{N}$ oraz wysokość $F(x_i)$, gdzie x_i to środek i -tego przedziału.

Przykładowo, dla $N = 10$ mamy:

- $\Delta x = 0.1$
- $x_1 = (1 - 0.5)\Delta x = 0.05$
 $x_2 = (2 - 0.5)\Delta x = 0.15$
...
 $x_{10} = (10 - 0.5)\Delta x = 0.95$
- $F(x_1) = \frac{4}{1+0.05^2} \approx 3.990024937655861$
 $F(x_2) = \frac{4}{1+0.15^2} \approx 3.911980440097800$
...
 $F(x_{10}) = \frac{4}{1+0.95^2} \approx 2.102496714848883$
- $\sum_{i=1}^{10} F(x_i) \Delta x \approx 3.142425985001098$



Program obliczający przybliżenie π należy zrównoleglić za pomocą OpenMP. Sprawdzić czas działania programu dla wartości $N \in \{10^6, 10^7, 10^8, 10^9, 10^{10}\}$ i dla liczby wątków równej: 1, 2, 4, 8, 16. Czasy działania wpisać do tabelki:

	10^6	10^7	10^8	10^9	10^{10}
1					
2					
4					
8					
16					

Zadanie 2

Dla danej liczby rzeczywistej x oznaczmy przez $\pi(x)$ liczbę liczb pierwszych nie większych od x . Przykładowo: $\pi(10) = 4$, $\pi(100) = 25$, $\pi(2^{10}) = 172$. Twoim zadaniem jest napisać program szukający wartości $\pi(2^n)$ dla jak największych n . Program ma działać nie dłużej niż jedną minutę.

Zadanie 3

Napisz program wyznaczający liczby pierwsze metodą sita Eratostenesa.

Krok 1 – napisz zwykłe sito Eratostenesa

Krok 2 – napisz tzw. segmented sieve

Krok 3 – zrównoleglij program i oblicz $\pi(2^{30})$

Porównaj uzyskane wyniki oraz czasy wykonania z tymi, które uzyskałeś w poprzednim zadaniu.

Zadanie 4

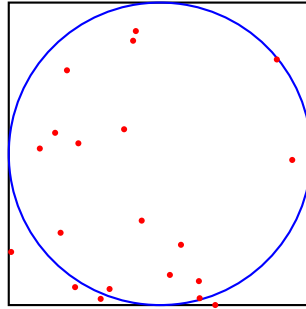
Dany jest zbiór $\{1, 2, 3, \dots, n\}$. Ile jego podzbiorów ma sumę elementów podzielnych przez 5? Odpowiedź dla $n = 10$ wynosi 208 (licząc też zbiór pusty). Znajdź wartość dla $n = 30$. Rozwiązanie ma generować wszystkie możliwe podzbiory i korzystać z OpenMP. Przewidywany czas działania programu – około 15 sekund.

Zadanie 5

Dany jest zbiór $\{1, 2, 3, \dots, n\}$. Ile jego permutacji nie ma żadnego punktu stałego? Odpowiedź dla $n = 5$ wynosi 44. Znajdź wartość dla $n = 11$. Rozwiązanie ma generować wszystkie możliwe permutacje i korzystać z OpenMP. Przewidywany czas działania programu – około 15 sekund.

Zadanie 6

Przybliżenie liczby π można uzyskać za pomocą algorytmu Monte Carlo. W kwadracie o boku długości 2 wpisano koło o promieniu $r = 1$. Algorytm polega na sukcesywnym losowaniu punktów należących do kwadratu.



Prawdopodobieństwo, że wylosowany punkt należy do koła wynosi:

$$P = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}$$

Dzieląc liczbę punktów w kole przez liczbę wszystkich wylosowanych punktów, otrzymujemy oszacowanie P , a stąd wartość π . Program należy zrównoleglić, pamiętając, aby każdy wątek korzystał z odrębnego generatora liczb pseudolosowych.

Zadanie 7

Zrównoleglić „naiwny” algorytm mnożenia dwóch macierzy A o wymiarach $m \times p$ oraz B o wymiarach $p \times n$:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pn} \end{pmatrix}$$

Wynikiem jest macierz C o wymiarach $m \times n$:

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

taka, że:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj}$$

Sprawdzić czas działania programu dla różnych wymiarów macierzy.

Czy czas działania zmieni się, gdy macierz B najpierw transponujemy? Wtedy:

$$c_{ij} = a_{i1}b_{j1}^T + a_{i2}b_{j2}^T + \cdots + a_{im}b_{jm}^T$$

gdzie b_{ij}^T oznacza element transponowanej macierzy B .

Przetestuj również użycie `#pragma omp for collapse(3)`