ZADANIE dla uczestników MiNI-Akademii
(metoda bisekcji)

A) Rozwiąż równanie $x^3 = 100 \cdot x \cdot \log_2 x$, wykorzystując metodę bisekcji ($x \in R$).
Opisz trzy kolejne kroki odpowiedniego algorytmu, podając przedziały, w których algorytm szuka rozwiązania oraz trzy kolejne rozwiązania przybliżone (przedział początkowy wybierz samodzielnie). Podaj ostateczne rozwiązanie, wyznaczone z dokładnością trzech miejsc po przecinku. Określ liczbę iteracji, jaka była potrzebna do uzyskania tego rozwiązania.

B) Załóżmy, że dane są dwa algorytmy: A_1 o złożoności czasowej n^3, oraz A_2 o złożoności czasowej $100 \cdot n \cdot \log_2 n$. Na podstawie rozwiązania z punktu A) sformułuj wnioski dotyczące porównania szybkości tych dwóch algorytmów zależnie od *n*.

Uwaga: Jeśli punkt A) zostanie rozwiązany za pomocą samodzielnie napisanego programu komputerowego, należy dołączyć do rozwiązania jego zapis (z komentarzami objaśniającymi główne kroki) lub precyzyjny opis zastosowanego algorytmu.

Rozwiązanie

A)
Część A należy rozwiązać omawianą na warsztatach metodą bisekcji – zastosowaną do wyznaczenia pierwiastków równania $x^3 - 100 \cdot x \cdot \log_2 x = 0$.
Analiza przebiegu dwóch funkcji występujących w tym równaniu wykazuje, że ma ono dwa pierwiastki: pierwszy – bliski jedynce, drugi – większy od 10.
Z punktu widzenia części B (analiza złożoności algorytmów) istotny jest drugi pierwiastek.

Obydwa pierwiastki można znaleźć, stosując opisany niżej algorytm iteracyjny.
Przyjęto w nim następujące oznaczenia:
f(x) – funkcja której pierwiastków szukamy; w zadaniu f(x) = $x^3 - 100 \cdot x \cdot \log_2 x$
a i b – odpowiednio lewy i prawy kraniec przeszukiwanego przedziału,
d – dokładność obliczeń,
N- maksymalna liczba iteracji.

<u>Algorytm</u>
1. Podaj a, b, d i N
2. i := 0
3. z_0 := a
4. Dopóki (|a-b| >d) oraz (i ≤ N) wykonuj
  4.1. i:=i+1
  4.2. p:=(a+b)/2
  4.3. Jeśli (f(a)·f(b)>0), to a:=p; w przeciwnym razie b:=p.
  4.4. z_i=p.
5. Zwróć z_i

Poszukując pierwszego pierwiastka, można przyjąć następujące wartości parametrów:
a=1, b=9, d=0,001, N=50.
Otrzyma się wówczas następujące rozwiązanie (przybliżone):

x = 1,007 (liczba iteracji: 14)

Poszukując drugiego pierwiastka, można przyjąć następujące wartości parametrów:
a=10, b=30, d=0,000001, N=50.
Otrzyma się wówczas następujące rozwiązanie (przybliżone):
x = 20,95 (liczba iteracji: 26)

B)
Na podstawie wartości drugiego pierwiastka oraz znajomości kształtu obydwu funkcji wnioskujemy, że dla $n \geq 21$ algorytm A_1 ma większą złożoność czasową niż A_2, czyli jest wolniejszy i mniej efektywny niż A_2. Dla $n < 21$ zachodzi sytuacja odwrotna. Ogólnie zatem, dla „dużych danych" (wysokie wartości n) algorytm A_2 jest bardziej efektywny niż A_1.