

Programowanie współbieżne

Lista 0

1. Dla zadanej liczby rzeczywistej a oraz dokładności ε można znaleźć pierwiastek trzeciego stopnia z a wyliczając kolejne przybliżenia x_i tego pierwiastka (metoda Newtona-Raphsona):

$$x_0 = a/3 \quad \text{dla } a > 1$$

$$x_0 = a \quad \text{dla } a \leq 1$$

$$x_{i+1} = x_i + (a/x_i^2 - x_i)/3$$

Dokładność jest osiągnięta, jeśli $|x_i^3 - a| \leq \varepsilon * |a|$.

Napisz efektywną (wykorzystującą rekursję ogonową) funkcję `root3: Double => Double`, która dla zadanej liczby a znajduje pierwiastek trzeciego stopnia z dokładnością $\varepsilon = 10^{-35}$,

np. `math.abs(root3(8.0) - 2.0) <= 1.0e-70`
`math.abs(root3(-8.0) + 2.0) <= 1.0e-70`