Practice Arena

Practice problems aimed to improve your coding skills.

The Bisection Method

LAB-PRAC-13_STRUC-NUM

The Bisection Method [20 marks]

Problem Statement

Let $f:\mathbb{R}\to\mathbb{R}$ be a continuous real-valued function. Then if there exist 2 points $a, b \in \mathbb{R}$ such that $f(a)f(b) < 0$, then there exists at least one root of $f(x) = 0$ in the interval (a, b). This can be formally proven using the intermediate value theorem (IVT).

In this problem, we will use the bisection method to find a root of a polynomial function using the bisection method. Suppose $f(a) > 0$ and $f(b) < 0$. Then the bisection method proceeds as follows

1. Find the midpoint of a & b. Call this mid point m. Then the following cases are possible
2. Case 1: If $f(m) = 0$, then we have found a root. We end the algorithm here itself.
3. Case 2: $f(m) < 0$, then applying the IVT again tells us that a root must lie between in the interval (a, m). Repeat the procedure on this new interval i.e. with the setting $a = a$ and $b = m$.
4. Case 3: $f(m) > 0$, then applying the IVT again tells us that a root must lie between in the interval (m, b). Repeat the procedure on this new interval i.e. with the setting $a = m$ and $b = b$.
5. Keep repeating the above procedure till we have $b - a < eps$ (remember that we always have b > a in our algorithm) where eps will be given to you in the input. When b and a are indeed strictly closer than eps, simply output $(a + b)/2$ as our approximation to the root.

We can mathematically show that this algorithm will always give us a number which is no farther than eps from a true root of the function. It can also be shown that the above algorithm will always stop in less than about $log\frac{1}{eps}$ steps.

The first line of the input will give you n, the degree of a polynomial, a, b and eps, all separated by a space. n will be a strictly positive integer whereas a, b, eps will be floating point numbers. Use float variables to store them and also use float variables in all your calculations. The second line will give you the n+1 coefficients of the polynomial starting from the zero-order coefficient and moving on to higher powers. All coefficients will be integers. For example, if n = 3, and the polynomial is

$$f(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d,$$

then we will give you the coefficients in the following order
d c b a

If we have given you invalid inputs, i.e. if $f(a) \cdot f(b) > 0$, then print "INVALID INPUT" in the output. If we have given you inputs such that $f(a) \cdot f(b) = 0$, then either a or b is a root. If both are roots then output the smaller root, correct to two decimal places using the %0.2f flag in printf, else if only one of a and b is a root then output that root, correct to two decimal places using the %0.2f flag in printf. In the general case, give the output correct to two decimal places using the %0.2f flag in printf.

Caution

1. Although we know that checking for equality with floating point numbers is not good, in order to simplify this question, we are asking you to check for exact equality when you check for

$f(m) == 0$.

2. Note that we will not always ensure that $f(a) > 0$ in the input. We may give you a case where $f(a) < 0$ and $f(b) > 0$. Even that is a valid case and you have to process it accordingly by applying the IVT.
3. Be careful that whereas eps will be a strictly positive floating point number, a, b can be zero or negative as well.
4. The coefficients of the function can be zero or negative as well.
5. Be careful about extra/missing lines and extra/missing spaces in your output.

EXAMPLE:
INPUT
3 1 5 0.1
-27 0 0 1

OUTPUT:
3.00

Grading Scheme:
Total marks: **[20 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

Start Solving! (/editor/practice/6259)