Brent's Method

The objective of this assignment is to write the **R** function `brent.r` that implements Brent's method for calculating the root of a function on a closed interval $[a, b]$. Brent's method converges very, very quickly. However, it turns out that you can program Brent's method very, very incorrectly, and it will still work, just not very efficiently.

If you read the code, you will see that during each iteration of the "while" loop, a new value of "s" is generated, either by the "inverse quadratic rule," the "secant rule," or the "bisection method." Then, if $f(s)$ has the same sign as $f(a)$, set b equal to s. Otherwise, set a equal to s. After that, a and b are updated so that $|f(a)| \geq |f(b)| \geq 0$.

Also, because the value of b changes each iteration, the value of δ also changes during each iteration. So, you cannot set the value of δ before the "while" loop.

The value of b is always the current "guess" for the root of f, and a is the "contrapoint." That is, $f(a)$ and $f(b)$ will always have different signs, and $|f(a)| \geq |f(b)| \geq 0$ so that b is always a better "guess" than a is. Geometrically, $f(b)$ is closer to the x-axis than $f(a)$ is.

Inputs: f, a, b, and a tolerance t

If $f(a)f(b) \geq 0$

Give a message to the user saying that $f(a)$ and $f(b)$ must have opposite signs

return "NA"

If $|f(a)| < |f(b)|$

swap a and b

$c \leftarrow a$

$mflag \leftarrow 1$

While (absolute distance between a and b is greater than t) **AND** (relative distance between a and b is greater than t),

If $(f(a) \neq f(c)$ AND $f(b) \neq f(c))$

$$s \leftarrow \frac{af(b)f(c)}{(f(a)-f(b))(f(a)-f(c))} + \cdots \\ \frac{bf(a)f(c)}{(f(b)-f(a))(f(b)-f(c))} + \frac{cf(a)f(b)}{(f(c)-f(a))(f(c)-f(b))}$$

Print "Inverse Quadratic Interpolation"

else

$s \leftarrow b - f(b)\frac{b-a}{f(b)-f(a)}$

Print "Secant Method"

If

$(s > \frac{3a+b}{4}$ **AND** $s > b)$ **OR**

$(s < \frac{3a+b}{4}$ **AND** $s < b)$ **OR**

$(mflag == 1$ **AND** $|s-b| \geq \frac{|b-c|}{2})$ **OR**

$(mflag == 0$ **AND** $|s-b| \geq \frac{|c-d|}{2})$ **OR**

$(mflag == 1$ **AND** $|b-c| < |\delta|)$ **OR**

$(mflag == 0$ **AND** $|c-d| < |\delta|)$, **then**

$s \leftarrow \frac{a+b}{2}$

$mflag \leftarrow 1$

Print "Bisection Method"

else

$mflag \leftarrow 0$

$d \leftarrow c$

$c \leftarrow b$

If $f(a)f(s) < 0$

$b \leftarrow s$

else

$a \leftarrow s$

If $|f(a)| < |f(b)|$

swap a and b

return b

Note that the c is the value of b during the previous iteration, and d is the value of b during the iteration before that. The program also calls for a value of δ. For this particular method, set

$$\delta = 2 \times \epsilon_{mach} \times |f(b)| + \frac{t}{2},$$

where ϵ_{mach} represents *machine epsilon*, and t represents the tolerance defined by the user. Note that δ needs to be recomputed during each iteration of the "while" loop because the value of b changes each time. The inputs to the function will be f, the function whose root we seek; a, the left endpoint of the interval; b, the right endpoint of the interval; and t, the tolerance defined by the user. Finally, continue the "while" loop until either the absolute error or the relative error is less than the tolerance.

Here are several examples. In both examples, your function should invoke the Inverse Quadratic Interpolation, the Secant Method, and the Bisection Method in the order that I have here. **Otherwise, you have not programmed the function correctly.**

```
> brent(my_sine,1,5,2^(-26))
Secant Method
Inverse Quadratic Interpolation
Secant Method
Secant Method
Bisection Method
Secant Method
Bisection Method
Secant Method
Bisection Method
Secant Method
Bisection Method
Secant Method
Bisection Method
Secant Method
Bisection Method
Secant Method
Secant Method
[1] 3.141593
> brent(my_tangent,0,5,2^(-26))
```

```
Secant Method.
Inverse Quadratic Interpolation.
Bisection Method.
Inverse Quadratic Interpolation.
Secant Method.
Secant Method.
Inverse Quadratic Interpolation.
Secant Method.
Secant Method.
[1] 1.732051
```