Elegant and Efficient Solution

for

Problem 4 of *Software Solutions*:

Different Neighbour

www.win.tue.nl/~wstomv

The Problem

$$a < b \ \wedge \ f(a) \neq f(b)$$

?

$$a \leq r < b \ \wedge \ f(r) \neq f(r+1)$$

Introducing a Fresh Variable

$$a < b \ \wedge \ f(a) \neq f(b)$$

$$\boxed{?}$$

$$a \leq r < s \leq b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \leq r < b \ \wedge \ f(r) \neq f(r+1)$$

Introducing a Loop

$$a < b \ \wedge \ f(a) \neq f(b)$$

?

invariant $a \le r < s \le b \ \wedge \ f(r) \neq f(s)$

variant function $s - (r + 1)$

loop while $s \neq r + 1$

?

end loop

$$a \le r < s \le b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \le r < b \ \wedge \ f(r) \neq f(r + 1)$$

Initialising the Loop Invariant

$$a < b \;\wedge\; f(a) \neq f(b)$$

$$r, s := a, b$$

invariant $a \leq r < s \leq b \;\wedge\; f(r) \neq f(s)$

variant function $s - (r + 1)$

loop while $s \neq r + 1$

?

end loop

$$a \leq r < s \leq b \;\wedge\; f(r) \neq f(s) \;\wedge\; s = r + 1$$

$$a \leq r < b \;\wedge\; f(r) \neq f(r + 1)$$

Refining the Loop Body

$$a < b \ \wedge \ f(a) \neq f(b)$$

$$r, s := a, b$$

invariant $a \leq r < s \leq b \ \wedge \ f(r) \neq f(s)$

variant function $s - (r + 1)$

loop while $s \neq r + 1$

$m := \boxed{?}$

if $\boxed{?} \rightarrow r := m$ [] $\boxed{?} \rightarrow s := m$ **fi**

end loop

$$a \leq r < s \leq b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \leq r < b \ \wedge \ f(r) \neq f(r + 1)$$

Determining the Condition on m

$$a < b \ \wedge \ f(a) \neq f(b)$$

$$r, s := a, b$$

invariant $a \leq r < s \leq b \ \wedge \ f(r) \neq f(s)$

variant function $s - (r + 1)$

loop while $s \neq r + 1$

$$m := \boxed{?}$$

$$r < m < s$$

if $\boxed{?} \rightarrow r := m$ [] $\boxed{?} \rightarrow s := m$ **fi**

end loop

$$a \leq r < s \leq b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \leq r < b \ \wedge \ f(r) \neq f(r + 1)$$

Choosing a Value for m

$$a < b \ \wedge \ f(a) \neq f(b)$$

$$r, s := a, b$$

$$\textbf{invariant} \ a \leq r < s \leq b \ \wedge \ f(r) \neq f(s)$$

$$\textbf{variant function} \ s - (r + 1)$$

$$\textbf{loop while} \ s \neq r + 1$$

$$m := (r + s) \ \textbf{div} \ 2$$

$$r < m < s$$

$$\textbf{if} \ \boxed{?} \rightarrow r := m \ [] \ \boxed{?} \rightarrow s := m \ \textbf{fi}$$

$$\textbf{end loop}$$

$$a \leq r < s \leq b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \leq r < b \ \wedge \ f(r) \neq f(r + 1)$$

Determining the Conditions for r, s

$$a < b \ \wedge \ f(a) \neq f(b)$$

$$r, s := a, b$$

invariant $a \leq r < s \leq b \ \wedge \ f(r) \neq f(s)$

variant function $s - (r + 1)$

loop while $s \neq r + 1$

$m := (r + s)$ **div** 2

$$r < m < s$$

if $f(m) \neq f(s) \rightarrow r := m$ [] $f(r) \neq f(m) \rightarrow s := m$ **fi**

end loop

$$a \leq r < s \leq b \ \wedge \ f(r) \neq f(s) \ \wedge \ s = r + 1$$

$$a \leq r < b \ \wedge \ f(r) \neq f(r + 1)$$

The Program Constructed, a.k.a. Binary Search

$r, s := a, b$

; **do** $s \neq r + 1 \rightarrow$

$m := (r + s)$ **div** 2

; **if** $f(m) \neq f(s) \rightarrow r := m$

[] $f(r) \neq f(m) \rightarrow s := m$

fi

od

Efficiency: constant memory and logarithmic time

$b - a = 10^9$ takes just 30 iterations; $b - a = 10^{18}$ takes 60

In the C programming language

```
r = a ; s = b ;
while ( s != r + 1 ) {
  m = ( r + s ) / 2 ;
  if    ( f(m) != f(s) )  r = m ;
  else /* f(r) != f(m) */ s = m ;
} /* end while */
```

Is this not beautiful?