# *Points on circle*

**Problem**

$N$ distinct points, numbered from 0 onwards, are located on a circle (in the rest of this problem all point numbers are taken $\mathbf{mod} N$). Point $i+1$ is the clockwise neighbor of point $i$. An integer array, $dist[0 \ldots N)$, is given such that $dist.i$ is the distance (along the circle) between points $i$ and $i+1$. Derive a program to determine whether four of these points form a rectangle.

We adopt the same notation used in *Programming in the 1990s* [1] and *Programming, The Derivation of Algorithms*[2]: The notation of function application is the "dot" notation with name of function, followed by arguments, each separated by a dot. The notation of quantified expressions has the operator followed by the bounded variables, then a colon followed by the range for the bounded variables and ended with a colon and the actual expression. So

$$(\sum k : i \leq k < j : x_k)$$

corresponds to the more classical mathematical notation $\sum_{k=i}^{j-1} x_k$.
For our derivation steps in predicate calculus we will use the following notation:

$$\begin{array}{ll} & A \\ = & \{\text{reason why A equals B}\} \\ & B \\ \leq & \{\text{reason why B is less than C}\} \\ & C \end{array}$$

We are asked to solve $S$ in

$|[$

$\quad$ **con** $N : \; int; \; \{N \geq 4\}$

$\quad\quad dist(i : 0 \leq i < N) : \; int; \; \{\forall i : 0 \leq i < N : dist.i > 0\}$

[1] Edward Cohen. *Programming in the 1990s, An Introduction to the Calculation of Programs*. Springer-Verlag, 1990

[2] A. Kaldewaij. *Programming, The Derivation of Algorithms*. Prentice Hall, 1990

**var** $r : bool$;
S
$\{r : r \equiv (\exists\, 4 \text{ points that form a rectangle})\}$
]|

Let's first develop a more manageable postcondition. Evidently four points that form a rectangle is equivalent to two pairs of diametral opposing points. We introduce a function for the set of all indices from point $x$ to point $y$ in clockwise direction along the circle:

$$
\begin{aligned}
&I : [0, \ldots, N) \to [0, \ldots, N) \to 2^{[0,\ldots,N)}\\
&I.x.y := \begin{cases} [x, \ldots, y) & , \; x \le y \\ [x, \ldots, N) \cup [0, \ldots, y) & , \; x > y \end{cases}
\end{aligned}
$$

Let $C$ be the circumference of the circle. We define function

$$
\begin{aligned}
&f : [0, \ldots, N) \to [0, \ldots, N) \to int\\
&f.x.y := C - 2(\textstyle\sum i : i \in I.x.y : dist.i)
\end{aligned}
$$

We want to find the number of diametral opposing pairs of points:

|[
**con** $N : \; int; \; \{N \ge 2\}$
$dist(i : 0 \le i < N) : \; int; \; \{\forall i : 0 \le i < N : dist.i > 0\}$
**var** $r : int$;
S
$\{r : r = (\#\, x, y : 0 \le x < N, \; 0 \le y < N : f.x.y = 0)\}$
]|

**Lemma 1.1.** *The function $f$ is increasing in its first argument and decreasing in its second argument.*

*Proof.* $f$ is increasing in its first argument:

$$
\begin{aligned}
& f.(x+1).y\\
=\; & \{\text{definition of } f\}\\
& C - 2(\textstyle\sum i : i \in I.(x+1).y : dist.i)\\
=\; & \{I.(x+1).y = I.x.y \setminus \{x\}\}\\
& C - 2((\textstyle\sum i : i \in I.x.y : dist.i) - dist.x)\\
=\; & \{\text{definition of } f\}\\
& f.x.y + 2dist.x\\
>\; & \{dist.x > 0\}\\
& f.x.y
\end{aligned}
$$

$f$ is decreasing in its second argument:

$$
\begin{array}{ll}
 & f.x.(y+1) \\
= & \{\text{definition of } f\} \\
 & C - 2(\sum i : i \in I.x.(y+1) : dist.i) \\
= & \{I.x.(y+1) = I.x.y \cup \{y\}\} \\
 & C - 2((\sum i : i \in I.x.y : dist.i) + dist.y) \\
= & \{\text{definition of } f\} \\
 & f.x.y - 2dist.y \\
< & \{dist.y > 0\} \\
 & f.x.y
\end{array}
$$

□

Looking at the postcondition

$$\{r : r = (\# \, x, y : 0 \le x < N, \ 0 \le y < N : f.x.y = 0)\}$$

we define the function

$$G.a.b = (\# \, x, y : a \le x < N, \ b \le y < N : f.x.y = 0)$$

and we will maintain the invariants:

$$
\begin{array}{lll}
P_0 & : & G.0.0 = r + G.a.b \\
P_1 & : & 0 \le a \le N \\
P_2 & : & 0 \le b \le N
\end{array}
$$

The initial values $r, a, b := 0, 0, 0$ satisfy the invariants and

$$a = N \vee b = N \Rightarrow G.a.b = 0 \Rightarrow r = G.0.0$$

establishes the postcondition, so we can stop when $a = N \vee b = N$.
So far we have

$\|[$
- **con** $N : \ int; \ \{N \ge 4\}$
  - $dist(i : 0 \le i < N) : \ int; \ \{\forall i : 0 \le i < N : dist.i > 0\}$
- **var** $a, b, r : int;$
- $a, b, r := 0, 0, 0;$
- **do** $a \ne N \wedge b \ne N$
  - S
- **od**
- $\{r : r = G.0.0\}$

$]\|$

We need to increment $a, b$ and maintain the invariants:

$$\begin{array}{cl} & G.a.b \\ = & \{\text{definition of } G\} \\ & (\#\, x,y : a \le x < N,\ b \le y < N : f.x.y = 0) \\ = & \{\text{range split } x = a\} \\ & G.(a+1).b + (\#y : b \le y < N : f.a.y = 0) \\ = & \{f \text{ is decreasing in second argument (1.1), and assume } f.a.b < 0\} \\ & G.(a+1).b \end{array}$$

so $f.a.b < 0 \Rightarrow G.a.b = G.(a+1).b$. Similarly

$$\begin{array}{cl} & G.a.b \\ = & \{\text{definition of } G\} \\ & (\#\, x,y : a \le x < N,\ b \le y < N : f.x.y = 0) \\ = & \{\text{range split } y = b\} \\ & G.a.(b+1) + (\#x : a \le y < N : f.x.b = 0) \\ = & \{f \text{ is increasing in second argument (1.1), and assume } f.a.b > 0\} \\ & G.a.(b+1) \end{array}$$

so $f.a.b > 0 \Rightarrow G.a.b = G.a.(b+1)$. Also for the case $f.a.b = 0$ we have

$$\begin{array}{cl} & r + G.a.b \\ = & \{\text{definition of } G\} \\ & r + (\#\, x,y : a \le x < N,\ b \le y < N : f.x.y = 0) \\ = & \{\text{range split } x = a\} \\ & r + G.(a+1).b + (\#y : b \le y < N : f.a.y = 0) \\ = & \{f \text{ is decreasing in second argument (1.1), and assume } f.a.b = 0\} \\ & (r+1) + G.(a+1).b \end{array}$$

Our program becomes

```
|[
    con N : int; {N ≥ 4}
        dist(i : 0 ≤ i < N) : int; {∀i : 0 ≤ i < N : dist.i > 0}
    var a, b, r : int;
    a, b, r := 0, 0, 0;
    do a ≠ N ∧ b ≠ N
      if
      □ f.a.b > 0 → b := b + 1
      □ f.a.b < 0 → a := a + 1
      □ f.a.b = 0 → a, r := a + 1, r + 1
      fi
    od
    {r : r = G.0.0}
]|
```

We cannot have $f$ in the program text so the last thing we have to do is eliminate $f$. We do this by introducing a new variable $c : int$ and maintaining the additional invariant $P_3 : c = f.a.b$. Lemma 1.1 already showed us the expressions for $f$ when the first or the second argument increase, so our final program looks like this[3]

[3] The program is bound by the function $2N - a - b$ so it is $O(N)$. The solution is an example of the slope search technique.

$|[$

**con** $N : \ int; \ \{N \geq 4\}$

$\quad dist(i : 0 \leq i < N) : \ int; \ \{\forall i : 0 \leq i < N : dist.i > 0\}$

**var** $a, b, c, r : int;$

$a, b, c, r := 0, 0, C, 0;$

**do** $a \neq N \wedge b \neq N$

**if**

$\Box \ c > 0 \rightarrow b, c := b + 1, c - 2dist.b$

$\Box \ c < 0 \rightarrow a, c := a + 1, c + 2dist.a$

$\Box \ c = 0 \rightarrow a, c, r := a + 1, 2dist.a, r + 1$

**fi**

**od**

$\{r : r = G.0.0\}$

$]|$

# *Bibliography*


Edward Cohen. *Programming in the 1990s, An Introduction to the Calculation of Programs*. Springer-Verlag, 1990.

A. Kaldewaij. *Programming, The Derivation of Algorithms*. Prentice Hall, 1990.