Example: Minimal-sum Section of Array (cont'd).

Proving program correctness requires showing the Hoare triple

pre-condition $\{P\}$ post-condition

pre-condition: $n > 0$ (length of array positive)

post-condition:

$$\left.\begin{array}{l} \forall i\ \forall j : 0 \le i \le j < n \rightarrow s = \min S_{ij} \end{array}\right\} \text{ s is a minimal sum.}$$

To prove correctness, we need loop-inv. & loop-variant for "while" loop.

loop-variant is easy: $E(\vec{x}) = n - k$
(K is counter that increase from 0 to n-1, and so n-k decreases from n-1 to 0).

loop invariant is not difficult (requires understanding of program):

$$\left.\begin{array}{l} inv1(s,k) := \forall i\ \forall j : (0 \le i \le j < K \rightarrow s = \min S_{ij}) \\ inv2(t,k) := \forall i : (0 \le i < K \rightarrow t = \min s_{i,K-1}) \end{array}\right\} \text{loop-inv.} = Inv1 \wedge Inv2.$$

$n > 0$

$Inv1(a[0], 1) \wedge Inv2(a[0], 1) \wedge (n \ge 1)$

$k := 1$

$Inv1(a[0], k) \wedge Inv2(a[0], k) \wedge (n - k \ge 0)$

$t := a[0]$

$Inv1(a[0], k) \wedge Inv2(t, k) \wedge (n - k \ge 0)$

$s := a[0]$

$Inv1(s, k) \wedge Inv2(t, k) \wedge (n - k \ge 0)$

while $(k \ne n)$ do { $t := \min(t + a[k], a[k])$

$s := \min(s, t)$

$k := k + 1$ }

$Inv1(s, k) \wedge Inv2(t, k) \wedge (n - k \ge 0) \wedge (k = n)$

$Inv1(s, n)$