Minsegsumtwodim

Just when I had posed "minsegsum" to my students, Jon Bentley's article [0] appeared in the CACM. The article mentioned the analogous problem in two dimensions and more or less suggested that it was a difficult one. As it mentioned no complexity results for the two-dimensional problem, both Jayadev Misra and I started to think about it and arrived independently at the same solution. Since the solution is complicated enough to present a problem of presentation, I decided to devote a note to it.

Remark about notation. By way of experiment I shall denote functional application by an infix period; it has the highest binding power of all operators and is (as usual) left-associative. By way of further experiment, I shall also use it for subscription. (End of Remark about notation.)

The functional specification for minsegsumtwodim is

|[M, N: int $\{M \geq 1 \wedge N \geq 1\}$
; $C(m, n: 0 \leq m < M \wedge 0 \leq n < N)$ array of int
; |[x: int
  ; minsegsumtwodim
   $\{R:\ x = (\underline{\mathrm{MIN}}\ m0, m1, n0, n1: 0 \leq m0 \leq m1 \leq M \wedge 0 \leq n0 \leq n1 \leq N:$
     $(\underline{S}\ m, n: m0 \leq m < m1 \wedge n0 \leq n < n1: C.m.n))\}$
  ]|
]|

in which (as usual) the constants of the environment are declared in the outer block.

To begin with we observe that $R \Rightarrow x \leq 0$ since the summation may be over an empty range $(m0 = m1 \vee n0 = n1)$. Furthermore we observe that R is not changed if we change the range for m0 and m1 into $0 \leq m0 < m1 \leq M$, since then the summation over the empty rectangle is still included. We adopt this change. (Condition $M \geq 1$ has been chosen, rather than $M \geq 0$, to make this change permissible; $N \geq 1$ has been chosen because, in general, our solution is most attractive for $N \geq M$.)

Our next step is to rewrite R in terms of nested MIN's. (At this stage the reader is invited to convince himself of the correctness of my rewriting and urged to shelve for the time being the question *why* R is rewritten *this* way.)

$$
\begin{aligned}
R:\ x = (&\underline{\mathrm{MIN}}\ m0: 0 \leq m0 < M: \\
&(\underline{\mathrm{MIN}}\ m1: m0 < m1 \leq M: \\
&\quad(\underline{\mathrm{MIN}}\ n1: 0 \leq n1 \leq N: MS.m0.m1.n1)))
\end{aligned}
$$

$$
\begin{aligned}
MS.m0.m1.n1 = \\
&(\underline{\mathrm{MIN}}\ n0: 0 \leq n0 \leq n1: (\underline{\mathrm{S}}\ n: n0 \leq n < n1: Q.m0.m1.n))
\end{aligned}
$$

$$Q.m0.m1.n = (\underline{\mathrm{S}}\ m: m0 \leq m < m1: C.m.n)$$

Lines 2 through 4 of the above describe the linear minsegsum "in the n-direction" with m0 and m1 as parameters: the m-direction has been "pushed to the sides", i.e. to the outer minimizations and the inner summation respectively.

We next derive —analogously to EWD897— two relations for MS .

$$MS.m0.m1.0 = 0 \qquad (0)$$

and for $0 \leq n1 < N$

$MS.m0.m1.(n1+1)$
$= \{\text{definition of MS}\}$
$(\underline{MIN}\ n0 : 0 \leq n0 \leq n1+1 : (\underline{S}\ n : n0 \leq n < n1+1 : Q.m0.m1.n))$
$= \{\text{for } n0 = n1+1 \text{, the summation yields } 0\}$
$(\underline{MIN}\ n0 : 0 \leq n0 \leq n1 : (\underline{S}\ n : n0 \leq n < n1+1 : Q.m0.m1.n))\ \underline{min}\ 0$
$= \{\text{isolation of last term of summation}\}$
$(\underline{MIN}\ n0 : 0 \leq n0 \leq n1 : (\underline{S}\ n : n0 \leq n < n1 : Q.m0.m1.n)$
$\qquad + Q.m0.m1.n1)\ \underline{min}\ 0$
$= \{Q.m0.m1.n1 \text{ does not depend on } n0 \text{ and addition then distributes over minimization}\}$
$((\underline{MIN}\ n0 : 0 \leq n0 \leq n1 : (\underline{S}\ n : n0 \leq n < n1 : Q.m0.m1.n))$
$\ + Q.m0.m1.n1)\ \underline{min}\ 0$
$= \{\text{definition of MS}\}$
$(MS.m0.m1.n1 + Q.m0.m1.n1)\ \underline{min}\ 0 \quad . \qquad (1)$

For Q we derive (directly from its definition)

$$Q.m0.m0.n1 = 0 \qquad (2)$$

$$Q.m0.(m1+1).n1 = Q.m0.m1.n1 + C.m1.n1 \quad . \qquad (3)$$

Our rewritten R tells us that for any m0,m1 combination we have to find the minimum of MS.m0.m1.n1 for all n1, while (1) gives a recurrence relation for that sequence of values. The snag is that that recurrence relation contains the term Q.m0.m1.n1, which -see (3)- satisfies a recurrence relation over m1. In order to exploit these two "orthogonal" recurrence relations to the fullest the program evaluates for each value of m0 the M-m0 recurrences (1) in synchrony. To this end a local array MSv(m: m0 < m ≤ M) is introduced, such that whenever MSv.m1 is adjusted, its value changes from MS.m0.m1.n1 to MS.m0.m1.(n1+1). Furthermore a local scalar Qv is introduced such that whenever Qv is adjusted, its value changes from Q.m0.m1.n1 to Q.m0.(m1+1).n1.

The annotated program is given below. In the initialization of x, x ≤ 0 is used; in its adjustment we use that <u>min</u> is associative.

Because we wish to use each MS-value as soon as it is computed, i.e. equals MS.m0.m1.(n1+1), line 2 of the rewritten R is rewritten once more:

$$0 \;\underline{\min}\; (\underline{\mathrm{MIN}}\ n1: 0 \le n1 < N: MS.m0.m1.(n1+1)) \quad ,$$

where the initial 0 can be taken out and taken care of by the initialization.

```
|[ m0: int ; x:=0 ; m0 := 0
; do m0 ≠ M →
    |[ MSv(m: m0< m ≤ M) array of int
    ; m1, n1 : int
    ; m1:= m0 ; do m1 ≠ M → m1:= m1+1 ; MSv.m1 := 0 od
    ; n1 :=0 {(A m: m0 < m ≤ M: MSv.m = MS.m0.m.n1)}
    ; do n1 ≠ N →
        |[ Qv: int ; Qv, m1 := 0, m0 {Qv = Q.m0.m1.n1}
        ; do m1 ≠ M →
              Qv, m1 := Qv + C.m1.n1 , m1 +1
              {Qv = Q.m0.m1.n1}
             ; MSv.m1 := (MSv.m1 + Qv) min 0
             ; x := x min MSv.m1
          od
        ]| ; n1:= n1 +1
      od
    ]| ; m0 := m0+1
  od
]|
```

* * *

<u>Side remark</u> Eventually, we came up with a program that works for $M=0 \vee N=0$ as well. We could have made our analysis also applicable in that case by defining the minimum of an empty set as "big enough" – 0, say – . (End of Side remark.)