Final Exam : Board Position

CSE 130: Algorithm Design

English Summary of Solution

Find the top-left and the bottom-right coordinates. From there, loop through all the values using a nested loop.

Pseudocode

```
PROMPT for board filename
READ board FROM filename

PROMPT for coordinates
done ← FALSE
WHILE not done
   GET column0, row0
   GET column1, row1
   IF    0 <= row0    < board.num_row AND
         0 <= column0 < board.num_col AND
         0 <= row1    < board.num_row AND
         0 <= column1 < board.num_col
      done ← TRUE
   ELSE
      PUT error message

columnMin ← MIN(column0, column1)
columnMax ← MAX(column0, column1)
rowMin ← MIN(row0, row1)
rowMax ← MAX(row0, row1)
ASSERT 0 ≤ columnMin ≤ columnMax ≤ board.num_col
ASSERT 0 ≤ rowMin    ≤ rowMax    ≤ board.num_row

score ← 0
FOR column ← columnMin ... columnMax
    FOR row ← rowMin ... rowMax
        ASSERT 0 ≤ column ≤ board.num_col
        ASSERT 0 ≤ row    ≤ board.num_row
        score ← score + board[column][row]

PUT score
```

Metrics

Efficiency

Reading data from the file is O(*n*) where there are *n* elements in the board.

Prompting for the coordinates and finding the min/max values is O(1), independent of the size of the board or the size of the rectangle.

The nested FOR loop is the following:

```
score ← 0
FOR column ← columnMin ... columnMax   # O(n1) where n1 is the width of the rect
    FOR row ← rowMin ... rowMax        # O(n1xn2) where n2 is the height of the rect
        score ← score + board[column][row]
```

Note that the rectangle can never be larger than the board so the overall efficiency is O(*n*).

Malleability

Configurable malleability because the board is read from a file and the coordinates come from the user.

Understandability

Obvious. All the variable names clearly describe what they do and how they are related to each other.

Quality

Asserts are in the pseudocode.

	0	1	2	3	4	5	6	7
0	12	15	19	17	15	14	12	10
1	14	16	21	20	18	15	15	17
2	19	22	25	22	20	19	20	22
3	20	24	26	24	20	19	21	25
4	18	20	23	20	17	19	22	27
5	16	18	22	19	20	21	26	29
6	14	16	21	18	22	25	28	31
7	17	18	19	20	24	28	30	32

Trace of the key part of the algorithm:

```
A  score ← 0
B  FOR column ← columnMin ... columnMax
C      FOR row ← rowMin ... rowMax
D          score ← score + board[column][row]
```

Line	columnMin/columnMax	rowMin/rowMax	column/row	board[row][column]	score
A	3,4	4,5	/	/	0
B	3,4	4,5	3,/	/	0
C	3,4	4,5	3,4	20	0
D	3,4	4,5	3,4	20	20
C	3,4	4,5	3,5	19	20
D	3,4	4,5	3,5	19	39
B	3,4	4,5	4,/	/	39
C	3,4	4,5	4,4	17	39
D	3,4	4,5	4,4	17	56
C	3,4	4,5	4,5	20	56
D	3,4	4,5	4,5	20	76