CS19002 PDS Lab, Spring 2007

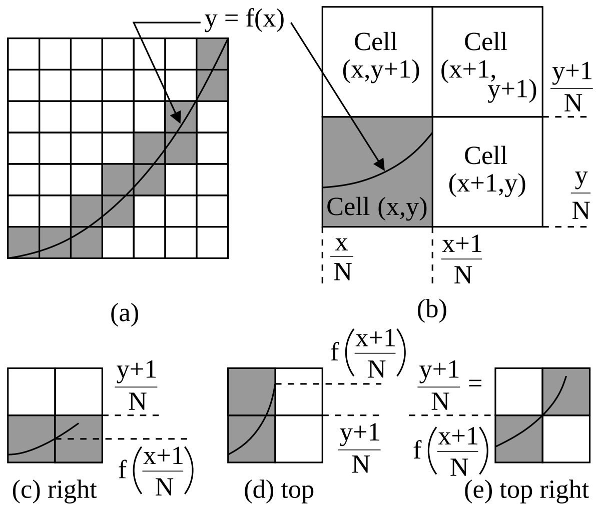
Lab test 2 (Section 5)

Total points: 30 | March 30, 2007 | Time: 14:30–16:30

For students with odd PC numbers

A treasure island of size 1 km $\times$ 1 km is divided into an $N \times N$ mesh of square cells of equal size. The cells are numbered (x, y) in the range $0 \leqslant x \leqslant N-1$ and $0 \leqslant y \leqslant N-1$. The bottom left cell has the number $(0, 0)$, whereas the top right cell has the number $(N-1, N-1)$. In the (x, y)-th cell, treasure worth $(x-y)^2$ million dollars is hidden.

You enter the island at the cell numbered $(0, 0)$, follow the curve $y = f(x)$, and leave the island at the cell numbered $(N-1, N-1)$. You collect all the treasure hidden in the cells encountered during the journey along the path $y = f(x)$. This situation is described in Part (a) of the adjacent figure. The visited cells are highlighted.



Part 1 Marks: **18**

In this part, you are asked to write a function that determines all the visited cells and the treasure collected at each visited cell. The function should also print the total number of cells visited and the total treasure collected during the entire journey. I now describe an algorithm to implement this function.

For simplicity, assume that the curve $y = f(x)$ is strictly increasing in the range $0 < x < 1$, and that $f(0) = 0$ and $f(1) = 1$. Suppose that you are currently in the cell (x, y). (Initially, $(x, y) = (0, 0)$.) The lower left corner of the island is taken as the origin, and distances are measured in units of kilometers. The boundaries of the current cell has x-coordinates x/N and $(x+1)/N$, and y-coordinates y/N and $(y+1)/N$. See Part (b) of the above figure.

Since the curve $y = f(x)$ is strictly increasing, the next cell to visit has to be one of the following: $(x+1, y)$ (movement to right), $(x, y+1)$ (movement to top), and $(x+1, y+1)$ (movement to top-right). In order to determine the next cell, compute the value $f((x+1)/N)$ at the right boundary of the current cell, and compare this value with the coordinate $(y+1)/N$. The three possibilities are illustrated in Parts (c), (d), and (e) of the above figure.

In order to complete Part 2 of the assignment, it is expedient to remember the path traversed. Use an $N \times N$ matrix M corresponding to the $N \times N$ array in the island. Before starting the journey, mark all entries in the matrix as "unvisited". Once you reach the cell (x, y), mark the element `M[y][x]` as "visited". (Notice that in the above figure, the x-coordinate grows horizontally towards right, and the y-coordinate grows vertically upward. On the other hand, the column numbering in a matrix grows horizontally towards right, whereas the row numbering grows vertically downward.)

Part 2 Marks: **12**

After the journey is complete, print the matrix M in order to show the path of your travel in the island. If `M[i][j]` is marked "visited", print the character `x`, otherwise print a space. Again notice that the y-coordinate in the plot grows vertically upward, whereas row numbers of a matrix grow downward. Print a row-major listing of M starting from the $(N-1)$-st row and ending in the 0-th row. This gives you a text plot of $y = f(x)$ in the range $0 \leqslant x \leqslant 1$.

Report the output of your program for $N = 40$ and $f(x) = 3x^2 - 2x^3$. Use integer arithmetic only in order to avoid floating-point errors.

For your convenience, a skeleton of the program is provided below.

```
#include <stdio.h>
#define N 40

void initMatrix ( int M[][N] ) ;
void collectTreasure ( int M[][N] ) ;
void plotPath ( int M[][N] ) ;

int main ()
{
   int M[N][N];

   initMatrix(M);        /* Initialize each cell of M to ‘unvisited’ */
   collectTreasure(M);   /* Part 1: Find the path in the island */
   plotPath(M);          /* Part 2: Plot the path */
}
```

Sample output

The following output corresponds to $N = 25$, $f(x) = (1 + 51x^2 - 49x^3)/10$, and the (x, y)-th cell storing treasure worth $x + y$ million dollars. The curve $y = f(x)$ in this example is neither strictly increasing nor strictly decreasing throughout the interval $0 < x < 1$. Your program can be easily modified to take care of such general functions.

```
Collecting $   2 million from cell ( 0, 2). Moving right.
Collecting $   3 million from cell ( 1, 2). Moving top.
Collecting $   4 million from cell ( 1, 3). Moving right.
Collecting $   5 million from cell ( 2, 3). Moving top.
...
Collecting $  31 million from cell (24, 7). Moving right.
*** Journey complete. Number of cells visited is 60.
*** Total treasure collected is worth $1596 million.
+-----------------------------------------------+
|                                               |
|                             x x x x x         |
|                           x x       x x       |
|                         x x           x       |
|                       x x             x x     |
|                     x x                 x     |
|                     x                   x x   |
|                   x x                     x   |
|                 x x                       x x |
|                 x                           x |
|               x x                           x |
|               x                             x |
|             x x                             x x
|             x                                 x
|           x x                                 x
|         x x                                   x
|         x                                     x
|       x x                                     |
|     x x                                       |
|   x x                                         |
| x x                                           |
x x                                             |
|                                               |
+-----------------------------------------------+
```