CPSC 535 8/29/2022

1. Assignment 1
   1. 1. Indicate the class Θ

For each of the following functions, indicate the class Θ(g(n)) the function belongs to. Apply all possible rules to simply g(n) as much as possible. Prove your assertions either using definitions or the limit theorem.

* + 1. a
    2. b. 2 n/2-1 + 3 n/3+1
  1. 2. Parallel matrix update of size 3x3

Given a matrix A of size 3x3, each cell (i.e. an element of the matrix) will be updated based on the values of (existing) immediate neighboring cells. This update must be done in parallel, this means A[0,0] will be updated in parallel with A[0,1], etc.. For a cell A[i,j], the update will be of the form:

A[i][j] = A[i-1][j] + A[i+1][j] + A[i][j-1] + A[i][j+1] - 4\*A[i][j]

Some of the terms may be missing if the element is on the border of the matrix: i=0, j=0, i=2, j=2.

For example A[0][0] = A[1][0] + A[0][1] - 4\*A[0][0].

In other terms, if A (0) is the initial matrix at time 0, then A (1) will be computed entirely based on the cells of A (0) at time 1, A (2) will be computed entirely based on the cells of A (1) at time 2, and so on.

To compute A (1) [0][0] one needs 3 operations (one addition, one subtraction, and one multiplication). You need to state the number of simple mathematical operations needed to compute all the cells at time 1. This value is obtained by adding the number of simple mathematical operations for each cell in the matrix.

Input: 3x3 Matrix  
Output: 3x3 matrix

**Design Matrix**:

Given a 3x3 Matrix L with update in the form of   
A[i][j] = A[i-1][j] + A[i+1][j] + A[i][j-1] + A[i][j+1] – 4\*A[i][j]

Procedure MatrixUpdate

Begin

Step 0:

Set A[0][0] as

A[(i),(j)] = min A[(i),(j)]

Step 1:

Add A[i-1] and A[i+1][j] and A[i][j-1] and A[i][j+1]

Set as L(m=1) = A[(i),(j)]

Set number of mathematical operations to with 3 for Step 1, 1 for step 2, 1 for step 3

Step 2:

multiply 4 to A[i][j]

L(m=2) = 4\*A[(i),(j)]

Step 3:

subtract L(m=1) from L(m=2)

Step 4: return new A[i][j]

L(m) = A[(i),(j)] = L(m=n)

End

* 1. 3. Parallel matrix update of size 3x3

Given a matrix A of size n x n, each cell (i.e. an element of the matrix) will be updated based on the values of (existing) immediate neighboring cells. This update must be done in parallel, this means A[0,0] will be updated in parallel with A[0,1], etc.. For a cell A[i,j], the update will be of the form:

A[i][j] = A[i-1][j] + A[i+1][j] + A[i][j-1] + A[i][j+1] - 4\*A[i][j]

Some of the terms may be missing if the element is on the border of the matrix: i=0, j=0, i=n-1, j=n-1.

For example A[0][0] = A[1][0] + A[0][1] - 4\*A[0][0].

In other terms, if A (0) is the initial matrix at time 0, then A (1) will be computed entirely based on the cells of A (0) at time 1, A (2) will be computed entirely based on the cells of A (1) at time 2, and so on.

To compute A (1) [0][0] one needs 3 operations (one addition, one subtraction, and one multiplication).

You need to state the number of simple mathematical operations needed to compute all the cells at time 1.

This value is obtained by adding the number of simple mathematical operations for each cell in the matrix.