CS3230 Tutorial 4 (Divide and Conquer) Sample Solution

February 17, 2017

1 Question 1

Let ModifiedMergesort be a mergesort modified to return the number of inversions in its input array $A[0 \ldots n-1]$ in addition to sorting it. Obviously, for an array of size 1, ModifiedMergesort($A[0]$) should return 0. Let i_{left} and i_{right} be the number of inversions returned by ModifiedMergesort($A[0 \ldots mid-1]$) and ModifiedMergesort($A[mid \ldots n-1]$), respectively, where mid is the index of the middle element in the input array $A[0 \ldots n-1]$.

The total number of inversions in $A[0 \ldots n-1]$ can then be computed as $i_{left} + i_{right} + i_{merge}$, where i_{merge}, the number of inversions involving elements from both halves of $A[0 \ldots n-1]$, is computed during the merging as follows. Let $A[i]$ and $A[j]$ be two elements from the left and right half of $A[0 \ldots n-1]$, respectively, that are compared during the merging. If $A[i] < A[j]$, we output $A[i]$ to the sorted list without incrementing i_{merge} because $A[i]$ cannot be a part of an inversion with any of the remaining elements in the second half, which are greater than $A[j]$. If, on the other hand, $A[i] > A[j]$, we output $A[j]$ and increment i_{merge} by $mid - i$, the number of remaining elements in the first half, because all those elements (and only they) form an inversion with $A[j]$.

2 Question 2

For $n > 1$, we can always place one L-tromino at the center of the $2^n \times 2^n$ chessboard with one missing square to reduce the problem to four subproblems of tiling $2^{n-1} \times 2^{n-1}$ boards, each with one missing square too. The orientation of this centrally placed piece is determined by the board's quarter with the missing square as shown by the example below in Figure 1.

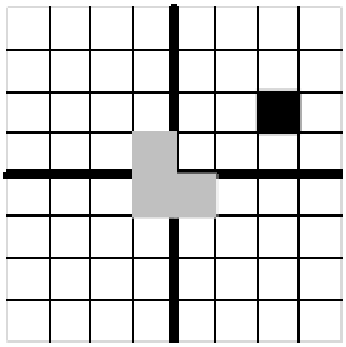

Figure 1: The orientation of the centrally placed tromino

Then each of the four smaller problems can be solved recursively until a trivial case of a 2×2 board with a missing square is reached.

3 Question 3

1. $A(n) = n - 1$

   Note that when $n = 1, 2$, the assertion is correct.

Assume for all $n \leq k$, $k \geq 2$ (strong math induction. **This step can NOT be omitted!!!**), we have $A(n) = n - 1$, when $n = k + 1$:

If $k + 1$ is even, we have:

$$A(k+1) = A(\frac{k+1}{2}) + A(\frac{k+1}{2}) + 1 = (\frac{k+1}{2} - 1) + (\frac{k+1}{2} - 1) + 1 = k + 1 - 1 = k$$

If $k + 1$ is odd, we have $\lfloor n/2 \rfloor = (n-1)/2$, $\lceil n/2 \rceil = (n+1)/2$.

$$A(k+1) = A(\frac{k}{2}) + A(\frac{k+2}{2}) + 1 = (\frac{k}{2} - 1) + (\frac{k+2}{2} - 1) + 1 = k - 1 + 1 = k$$

2. $B(n) = \lfloor \log n \rfloor + 1$

   Note that when $n = 1, 2$, the assertion is correct.

   Assume for all $n \leq k$, $k \geq 2$, we have $B(n) = \lfloor \log n \rfloor + 1$, when $n = k + 1$:

   If $k + 1$ is even, we have:

$$B(k+1) = B(\frac{k+1}{2}) + 1 = \lfloor \log \frac{k+1}{2} \rfloor + 1 + 1 = \lfloor \log \frac{k+1}{2} + 1 \rfloor + 1 = \lfloor \log(k+1) \rfloor + 1$$

   If $k + 1$ is odd, we have:

$$B(k+1) = B(\frac{k}{2}) + 1 = \lfloor \log \frac{k}{2} \rfloor + 1 + 1 = \lfloor \log \frac{k}{2} + 1 \rfloor + 1 = \lfloor \log k \rfloor + 1$$

   Let's assume $\lfloor \log k \rfloor = r \geq 0$, then we have $2^r \leq k < 2^{r+1}$, then $2^r + 1 \leq k + 1 < 2^{r+1} + 1$.

   On the other hand, $k + 1$ is odd, but $2^{r+1} + 1$ is also odd, we must have:

$$2^r < k + 1 < 2^{r+1}$$

   Thus,

$$r < \log(k+1) < r + 1 \Leftrightarrow \lfloor \log(k+1) \rfloor = r = \lfloor \log k \rfloor$$

   Finally, we have:

$$B(k+1) = \lfloor \log k \rfloor + 1 = \lfloor \log(k+1) \rfloor + 1$$

(Note: there also exists a more elegant solution to this question - writing n in binary representation.)

4 Question 4

Assume that days are numbered from 1 to 366 (Feb 29 is day 60), start with a question such as "is your birthday's number > 183". This is really a binary search problem. The largest number of questions that may be required is 9 ($= \lceil \log_2 366 \rceil$).

5 Question 5

Additional Assumption: can only break one bar at a time.

The answer is $nm - 1$.

Since only one bar can be broken at a time, any break increases the number of pieces by 1. Hence, $nm-1$ breaks are needed to get from a single n-by-m piece to nm one-by-one pieces, which is obtained by any sequence of $nm-1$ allowed breaks. (The same argument can be made more formally by mathematical induction.)

(**Optional**) Assume that we are allowed to pile the bars up and break many of them together at once, the minimal number of breaks required should be: $\lceil \log m \rceil + \lceil \log n \rceil$. Of course, we need to make sure the chocolate bars are not too thick - otherwise we can hardly break many of them together!

6 Question 6

Additional Assumption: all the tree nodes are distinct.

```
Tree_recover(T_pre, T_in, T_post)
  if T_pre contains 1 node then
     return that node
  fi
  find T, TL_pre, TL_in, TL_post, TR_pre, TR_in, TR_post
  TL <- Tree_recover(TL_pre, TL_in, TL_post)
  TR <- Tree_recover(TR_pre, TR_in, TR_post)
  return
         T
       /   \
     TL     TR
```

Further Discussion: The tree can never be recovered from a single list without prior assumptions on the tree structure. If we assume the binary tree to be recovered is always a full binary tree, then we can recover the tree from any two of the above three lists. If we assume the binary tree to be recovered can be an arbitrary binary tree (not necessarily full binary tree), then we can recover the tree from the following two combinations: `preorder` + `inorder`, or `postorder` + `inorder`.

7 Question 7

For the matrices given, Strassen's algorithm yields the following:

$$C = \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

where

$A_{00} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}$, $A_{01} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$, $A_{10} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}$, $A_{11} = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}$,

$B_{00} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}$, $B_{01} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}$, $B_{10} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$, $B_{11} = \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}$

Therefore,

$M_1 = (A_{00} + A_{11})(B_{00} + B_{11}) = \begin{bmatrix} 4 & 0 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix}$,

$M_2 = (A_{10} + A_{11})B_{00} = \begin{bmatrix} 3 & 1 \\ 7 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix}$,

$M_3 = A_{00}(B_{01} - B_{11}) = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -5 & 4 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix}$,

$M_4 = A_{11}(B_{10} - B_{00}) = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}$,

$M_5 = (A_{00} + A_{01})B_{11} = \begin{bmatrix} 3 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix}$,

$M_6 = (A_{10} - A_{00})(B_{00} + B_{01}) = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix}$,

$M_7 = (A_{01} - A_{11})(B_{10} + B_{11}) = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix}$

Accordingly,

$C_{00} = M_1 + M_4 - M_5 + M_7 = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} + \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$,

$C_{01} = M_3 + M_5 = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix} + \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 1 & 9 \end{bmatrix}$,

$C_{10} = M_2 + M_4 = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 1 \\ 5 & 8 \end{bmatrix}$,

$C_{11} = M_1 + M_3 - M_2 + M_6 = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix} - \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ 7 & 7 \end{bmatrix}$.

That is,

$$C = \begin{bmatrix} 5 & 4 & 7 & 3 \\ 4 & 5 & 1 & 9 \\ 8 & 1 & 3 & 7 \\ 5 & 8 & 7 & 7 \end{bmatrix}$$

Any bugs and typos, please report to Roger Zimmermann (rogerz@comp.nus.edu.sg).