Data Structures and Algorithms | January 28, 2017
Pennsylvania State University | CMPSC 465
Professors Adam Smith & Piotr Berman | Spring 2017

Homework 3 – Due Friday, February 3, 2017

Reminders

- Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. If you worked alone, write "Collaborators: None." Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

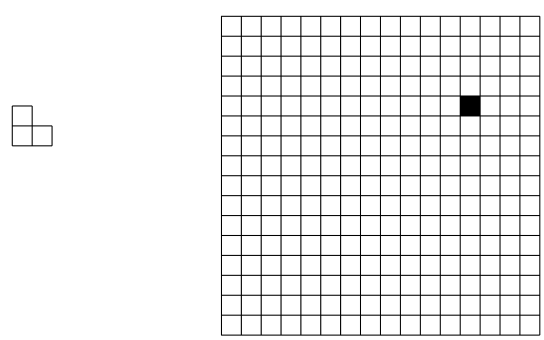
Problems to be handed in

1. **(Carpets)** You finally graduated from Penn State and took a job that allowed you to afford your very own apartment. Congratulations!

   The living space in your new apartment is a $2^k \times 2^k$-foot square (which you mentally divide into 2^{2k} cells, each 1 foot square). Bizarrely, exactly one cell of the floor is covered by carpet, and the rest is bare cement (Figure 1 (b)).

   Fortunately, a friend has given you a pile of $\frac{2^{2k}-1}{3}$ carpet pieces. Each carpet piece is L-shaped, formed by three 1-by-1 adjacent squares (Figure 1(a)). Each piece covers 3 cells, so you have just enough carpet to cover the whole cement part of the floor, if the pieces can fit.

   Your job is to find a way to lay the pieces so they cover the cement. The carpet should cover all cells except the covered one with no overlaps. You may not cut the carpet pieces—you have to use them as they are—but it is fine to rotate them.



(a) (b)

Figure 1: (a) A carpet piece in one of its 4 possible orientations. (b) A 16×16 floor with one covered cell.

Submit your solution to parts (a)-(c) as a PDF on Canvas, and your solution to (d) on Vocareum.

(a) Design a divide-and-conquer algorithm for this problem. The inputs are $n = 2^k$ (which determines the size of the room) and the coordinates $(x, y) \in \{1, ..., n\} \times \{1, ..., n\}$ of the missing square. You may assume n is an integer power of 2. The output should be a list of triples, where each triple describes the position of one of the carpet pieces you will put down.

First, explain your algorithm concisely in English (feel free to use pictures). Second, specify your algorithm using clear pseudocode or readable Python code.

(b) Prove that your algorithm is correct.

(c) Give a recurrence for the worst case running time of your algorithm in terms of n and solve it. How long does your algorithm take as a function of n?

(d) Implement your algorithm on Vocareum.

2. **(Recurrences)** [Submit your answers as PDF on Canvas.]

(a) For each of the following algorithms, write a recurrence relation that best describes the running time of the algorithm. You don't need to prove the algorithm is correct or solve the recurrence; just write it down.

Algorithm 1: FindMax(A, ℓ, r)

Input: A is an array of real numbers, indexed from 1 to n; $\ell, r \in \{1, ..., n\}$ satisfy $\ell \leq r$

if $\ell = r$ **then**
    return $A[\ell]$
else if $r - \ell = 1$ **then**
    return $\max(A[\ell], A[r])$
else
    $mid = \frac{\ell+r}{2}$;
    return max(FindMax(ℓ, mid),FindMax$(mid + 1, r)$)

To write a recurrence analyze the next algorithm, it is helpful to know that one can add and subtract two matrices (of the same dimensions) in time proportional to the number of entries in each of the matrices.

Algorithm 2: MatrixMultiply(A,B)

Input: A, B are $n \times n$ square matrices

if $n = 1$ **then**
    return $A \times B$
else
    Partition A into A^{11}, A^{12}, A^{21}, A^{22}
    Partition B into B^{11}, B^{12}, B^{21}, B^{22}
    // Where each is a quarter of the original matrices ($m \times m$ where $m = n/2$)
    $P_1 \leftarrow$ MatrixMultiply(A^{11}, $B^{12} - B^{22}$)
    $P_2 \leftarrow$ MatrixMultiply($A^{11} + A^{12}$, B^{22})
    $P_3 \leftarrow$ MatrixMultiply($A^{21} + A^{22}$, B^{22})
    $P_4 \leftarrow$ MatrixMultiply(A^{22}, $B^{21} - B^{11}$)
    $P_5 \leftarrow$ MatrixMultiply($A^{11} + A^{22}$, $B^{11} + B^{22}$)
    $P_6 \leftarrow$ MatrixMultiply($A^{12} - A^{22}$, $B^{21} + B^{22}$)
    $P_7 \leftarrow$ MatrixMultiply($A^{11} - A^{21}$, $B^{11} - B^{12}$)
    $C^{11} \leftarrow P_5 + P_4 - P_2 + P_6$
    $C^{12} \leftarrow P_1 + P_2$
    $C^{21} \leftarrow P_3 + P_4$
    $C^{22} \leftarrow P_1 + P_5 - P_3 - P_7$
    Combine C^{11}, C^{12}, C^{21}, and C^{22} into $n \times n$ matrix C
    return C

(b) For each of the following recurrences, give a one-line answer to the following questions: When you construct a recursion tree of the recurrence relation,

A. how many leaves does it have?
B. what is the height of the tree?
C. how many nodes does it have?

i. $T(n) = 4T\left(\frac{n}{2}\right) + n^2$
ii. $T(n) = T(n-1) + n$
iii. $T(n) = 3T\left(\frac{n}{5}\right) + n^3$
iv. $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + \sqrt{n}$

You may assume that $T(n) = 1$ for $n \leq 1$

(c) For the last recurrence (that is, $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + \sqrt{n}$), use the substitution method to prove that $T(n) = O(n)$. You may again assume that $T(n) = 1$ for $n \leq 1$.