Engineering and Applied Science Programs for Professionals
Whiting School of Engineering
Johns Hopkins University
Foundations of Algorithms
Homework 2
Due at the end of Module 4

Total Points 100/100

Collaboration groups will be set up in Blackboard by the end of the week. Make sure your group starts an individual thread for each collaborative problem and subproblem. You are required to participate in each of the collaborative problem and subproblem.

Self-Study Problems

All of the following problems come from the textbook and have solutions posted on the web at http://mitpress.mit.edu/algorithms.

You are permitted to use this site to examine solutions for these problems as a means of self-checking your solutions. These problems will not be graded.

Problems: 3.1-2, 4.2-4, 4.4-6, 17.1-3, 17.2-2, 17.2-3.

Problems for Grading

Determine if the statement is true or false for Problems 1, 2 and 3. If the statement is true or false, show which of the Master Theorem Cases applies. Show your work for the Case that proves the solution provided or the correct solution. If the Master Theorem cannot be used to show that the solution is true or false show each of the three cases and state that the Master Theorem cannot be used to solve the problem.

Your answers should be in the format shown on page 74 Second Edition or page 94 Third Edition.

Recall that **master theorem** is as follows:
Let $a \geq 1$ and $b > 1$ be constants and $f(n)$ be a function. Let $T(n)$ be defined on the nonnegative integers by the following recurrence

$$T(n) = aT(n/b) + f(n)$$

Notice that here n/b can be interpreted as either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows

Case 1 Recall, if there exists a constant $\epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$ then $T(n) = \Theta(n^{\log_b a})$.

Case 2 Recall, if there exists an integer $k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \log^k n)$ then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$. *Note: This formula of Case 2 is more general than Theorem 4.1, and it is given in Exercise 4.4-2 of the Second Edition and in Exercise 4.6-2 of the Third Edition. $f(n)$ is within a polylog factor of $n^{\log_b a}$, but not smaller. Polylog is described in Chapter 3 under Logarithms.* Intuitively the cost is $n^{\log_b a} \lg^k n$ at each level and there are $\Theta(\lg n)$ levels. For a simple case, $k = 0 \Rightarrow f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \lg n)$.

Case 3 Recall, if there exists a constant $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \epsilon})$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$, then $T(n) = \Theta(f(n))$.

1. **Problem 1 Chapter 4**
   10 Points

   T F The solution to the recurrence $T(n) = 3T(n/2) + n$ is $\Theta(n \ln n)$.

2. **Problem 2 Chapter 4**
   10 Points

   T F The solution to the recurrence $T(n) = T(\sqrt{n}) + 1$ is $\Theta(n^2)$.

3. **Problem 3 Chapter 4**
   10 Points

   T F The solution to the recurrence $T(n) = 2T(n/3 + 1) + n$ is $\Theta(n \ln n)$.

4. **Problem 4 Chapter 4 Parts a and b *Note this is a Collaborative Problem***
20 Points Total 10 Points Each

(a) Argue that the solution to the recurrence $T(n) = T(n/3) + T(2n/3) + cn$, where c is a constant, is $\Omega(n\lg n)$ by appealing to a recursion tree.

(b) Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.

5. **Problem 5 Chapter 17 Parts a, b *Note this is a Collaborative Problem***
30 Points Total Part (a) 5 Points, Part (b) 25 Points

In this problem we consider two stacks A and B manipulated using the following operations (n denotes the size of A and m the size of B):

- $PushA(x)$: Push element x on stack A.
- $PushB(x)$: Push element x on stack B.
- $MultiPopA(k)$: Pop $\min\{k, n\}$ elements from A.
- $MultiPopB(k)$: Pop $\min\{k, m\}$ elements from B.
- $Transfer(k)$: Repeatedly pop an element from A and push it on B, until either k elements have been moved or A is empty.

Assume that A and B are implemented using doubly-linked lists such that $PushA$ and $PushB$, as well as a single pop from A or B, can be performed in $O(1)$ time worst-case.

(a) What is the worst-case running time of the operations $MultiPopA$, $MultiPopB$ and $Transfer$?

(b) Define a potential function $\Phi(n, m)$ and use it to prove that the operations have amortized running time $O(1)$.