CSCI 705 Algorithms
Homework 2
Due 2/22/2024

Saad Mneimneh
Computer Science
Hunter College of CUNY

Readings
Based on Lectures 3, 4, and 5 and their assigned readings (see course website).

Problem 1
(Optional for fun, you might want to work on a subset of these)

Rank the functions by order of growth, where $f_1(n) = \Omega(f_2(n)), f_2(n) = \Omega(f_3(n)), \ldots, f_{29}(n) = \Omega(f_{30}(n))$. Use Θ instead of Ω if $f_i(n) = \Theta(f_{i+1}(n))$. Note that $\log^* n$ is the number of times the logarithm function must be applied until the result is less or equal to 1.

n^2	$n^{2\lg\lg n}$	$n^2 + 2^{100}n$	$\lfloor n \rfloor$	n^n	2^{2n}
$(\frac{3}{2})^n$	$(\frac{2}{3})^n$	$n^{\log_8 n}$	$(\lg n)!$	100^{100}	$(1/n)^{1/\lg n}$
$\ln\ln n$	$2^{\lg^* n}$	$n2^n$	$3(n!)$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$\sum_{k=1}^n k$	$(n+1)!$	$\sqrt{\lg n}$
$\lg(\lg^* n)$	$\lg^*(\lg n)$	n	2^n	$n\lg n$	$2^{2^{n+1}}$

Hint: Sometimes to compare two functions, it helps to compare their logarithms.

Problem 2: Looking into Quicksort

The analysis of randomized Quicksort relied on that all elements are distinct. We assume that the partitioning algorithm is the one presented in class, where $A[r]$ is the pivot:

RandPartition(A, p, r)
  $i \leftarrow$ random(p, r)
  swap $A[r] \leftrightarrow A[i]$
  return Partition(A, p, r) (see Partition in lecture notes)

(a) Explain why the assumption about elements being distinct is needed.

(b) Suggest a way to change the partitioning algorithm to overcome the problem of the multiplicity of elements.

(c) Assume that elements may not be distinct, but that every element cannot appear more then c times, where c is a constant. How does that affect the asymptotic running time of Quicksort?

Problem 3

Consider the following recurrence:

$$T(n) = 4T(n/2) + \Theta(n^2/\log n)$$

(a) Show that $T(n) = O(n^2 \log n)$ and that $T(n) = \Omega(n^2)$.

(b) Based on the above, make a guess for $T(n) = \Theta(n^2 \log\log n)$ and prove it using the substitution method.

(c) Using the fact that $1+1/2+1/3+\ldots+1/n = \Theta(\log n)$, use the recursive tree method to show that $T(n) = \Theta(n^2 \log\log n)$.

Problem 4

Give asymptotic upper and lower bounds for $T(n)$ which are as tight as possible. Assume that $T(n)$ is constant for $n \leq n_0$, where n_0 is a constant. For most of these, you can use the Master method. If not, find other ways such as guessing the answer and verifying it using the substitution method.

(a) $T(n) = 6T(n/3) + n^3$

(b) $T(n) = 6T(n/3) + n$

(c) $T(n) = 9T(n/3) + n^2$

(d) $T(n) = 8T(n/2) + n^3 \lg^2 n$

(e) $T(n) = 10T(n/3) + n^2\sqrt{n}$

(f) $T(n) = T(n/3) + 2T(n/4) + n$

(g) $T(n) = T(n^{1/3}) + \lg n$
Hint: change of variable.

(h) $T(n) = 3T(n-1) + n^3$
Hint: Guess from recursive tree and verify, or work directly with a sum.

(i) $T(n) = T(\lg n) + 1$
Hint: Review the definition of $\log^*$, and verify your guess by substitution method.

(j) $T(n) = T(n/4) + \sqrt{n}$

(k) $T(n) = T(n/2 + \sqrt{n}) + 1$
Hint: use the idea of lower and upper bounding this by two recurrences that are more friendly.